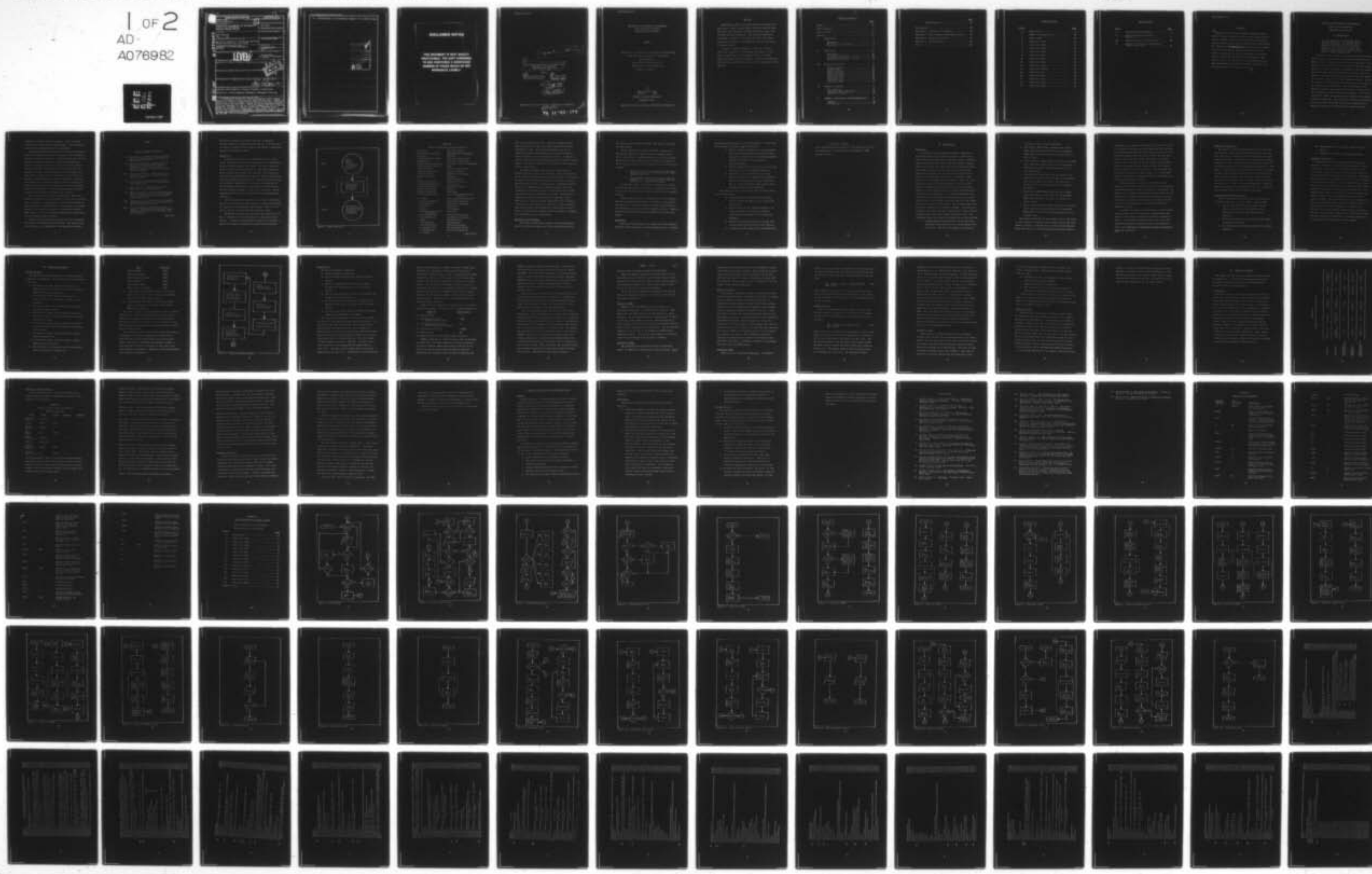
AD-A076 982 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 5/10
INTERACTIVE PROGRAMMING FOR MANAGEMENT AIDS TO DECISION-MAKING:--ETC(U)
SEP 79 P HAMILTON
UNCLASSIFIED AFIT/GSM/SM/79S-6 NL

1 OF 2
AD
A076982



AD A076982

DDC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GSM/SM/79S-6	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INTERACTIVE PROGRAMMING FOR MANAGEMENT AIDS TO DECISION-MAKING: QUEUEING THEORY APPLIED		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Paul Hamilton Capt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Department of Systems Management AFIT/EN		12. REPORT DATE September 1979
		13. NUMBER OF PAGES 127
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) LEVEL IV		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

DDC
NOV 7 1979
E

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17

JOSEPH P. HIPPS, Major, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Computer Aided Diagnosis; Computer Programs; Computerized Simulation; Decision-Making; Management; Management Training

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
Quantitative models for decision-making are useful only if they are available to those who must make the decisions. QUEUE1, an interactive computer program which incorporates six basic queueing theory models, was developed. QUEUE1 and the accompanying instructions were designed for use by mid-level managers who do not have an extensive knowledge of computer operations or modeling theory. QUEUE1 was tested on five individuals. Test results lead to the conclusion that interactive programming techniques can present quanti-

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. tative models to non-technical managers in a useable format.

Accession For	
NTIS GRA&I	✓
DDC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	23 CP

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

AFIT/GSM/SW/79S-6

9 Master's thesis,

6

INTERACTIVE PROGRAMMING FOR MANAGEMENT
AIDS TO DECISION-MAKING:
QUEUEING THEORY APPLIED.

THESIS

14 AFIT/GSM/SM/79S-6

10 Paul Hamilton
~~Captain USAF~~

11 Sep 79

12 122

Approved for public release: distribution unlimited

012225

79 11 05 172

AFIT/GSM/SM/79S-6

INTERACTIVE PROGRAMMING FOR MANAGEMENT AIDS TO DECISION-MAKING: QUEUEING THEORY APPLIED

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Paul Hamilton, B.S.
Captain USAF

Graduate Systems Management

September 1979

Preface

Quantitative aids to decision-making have existed for many years. Their complex nature, however, prevents many mid-level and lower-level managers from using these aids in their routine activities. This research represents an attempt to develop a technique that will make these aids more accessible to those managers who do not employ a research staff.

I wish to thank Dr. Edward J. Dunne, Jr. for his invaluable assistance in preparing this report. I also wish to express my love and appreciation to my wife, Charlotte, whose infinite patience, constant encouragement, and hard work contributed so much to the completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Quantitative models for decision-making are useful only if they are available to those who must make the decisions. QUEUE1, an interactive computer program which incorporates six basic queueing theory models, was developed. QUEUE1 and ~~the accompanying~~ instructions were designed for use by mid-level managers who do not have an extensive knowledge of computer operations or modeling theory. QUEUE1 was tested on five individuals. Test results lead to the conclusion that interactive programming techniques can present quantitative models to non-technical managers in a useable format.

INTERACTIVE PROGRAMMING FOR MANAGEMENT

AIDS TO DECISION-MAKING:

QUEUEING THEORY APPLIED

I. INTRODUCTION

> "... the computer, for those who understand anything about it, is emancipation for the individual. The purpose of the computer is to enable us not to spend time on 'controls' but to use the time for tasks that require perception, imagination, human relations, and creativity..."
> Peter F. Drucker (Ref 10:259)

A review of the literature concerning management reveals that the computer has entered the manager's domain and is there to stay. Indeed, the comment at the beginning of this chapter is echoed in one form or another in virtually every management journal and textbook published in the last decade. Still, when one examines the management theories and techniques in this same literature, the computer does not appear as a primary component of current management models. Rather, the computer is relegated to the ancillary role of an aid to the management of information. What is important is the growing concern by the management community over the effective integration between manager and computer. That is not to say that current management theory understates the power of modern technology. The concern is, in part, a realization that the complex technology continually outdistances the manager's ability to utilize the full potential of the

computer as an aid to decision-making. It can be demonstrated that this concern is well founded. First, however, the manager's viewpoint should be considered.

An informed manager realizes that even routine decisions cannot be formulated solely on the basis of "gut feelings," intuition, and prior practices. Effective decisions are the result of structured analysis of pertinent information. Analysis techniques should be consistent with the problem being considered. "The field of operations research, or management science, is concerned with the development and application of quantitative techniques to the solution of problems faced by managers of public and private organizations" (Ref 6:2). Modern management theory recognizes the need for quantitative decision-making in routine management efforts (Ref 24:145). A representative paradigm for quantitative decision-making is reproduced in Table I.

The accomplishment of Problem Formulation, Model Construction, Data Collection, and Model Solution are of concern here. A manager considering even routine problems might require the aid of a mathematician, a statistician, a research analyst and a computer programmer to effectively complete these steps. It is a fortunate manager who has ready access to such valuable human resources.

Clearly, many routine problem situations do not warrant the expense of providing these resources. The major intent of this thesis is to demonstrate that quantitative models

TABLE I

A Decision-Making Paradigm

I. Recognition of a Need. (The perception that some action needs to be taken, or perhaps taken better.)

II. Problem Formulation. (Translation of the perceived need into an explicit statement of both the need and the criteria by which the problem solution is to be judged.)

III. Model Construction. (Construction of a mathamatical replica or representation of the problem.)

IV. Data Collection. (The specific inputs to the model which reflect actual problem conditions.)

V. Model Solution. (Manipulation of the input data to produce results.)

VI. Model Validation and Sensitivity Analysis. (Testing model results to ensure validity and the implications of errors in estimating input data.)

VII. Interpretation of Results and Implications. (Broad reexamination of problem criteria in light of model results.)

VIII. Decision Making, Implementation, and Control. (Behavioral and technical change requirements in both short-run and long-run conditions.)

(Ref 6:3)

for decision-making can be made available to managers through interactive computer programming in a form that does not require technical proficiency in the fields of Operations Research and Computer Science.

Background

A quantitative model is a representation of a "real-world" system which is designed to describe the relationship between specified inputs and outputs. It can be viewed as an open system as shown in Figure 1. The input represents known variables which define the parameters and conditions upon which the model operates to produce desired information about the system being represented. This desired information is the output. Managers who lack extensive background in the principles of mathematics underlying the model may not be able to apply the model to practical situations. Inputs entered in error result in outputs that are misleading and confusing.

Since World War II, analysts in the field of Operations Research have developed a great number of quantitative models for application in government and industry. Table II is a list of some of the types that are available today.

The models listed in Table II often rely heavily on the use of computers to perform lengthy and complicated calculations. In general, managers do not possess the ability to perform the required mathematical manipulations manually.

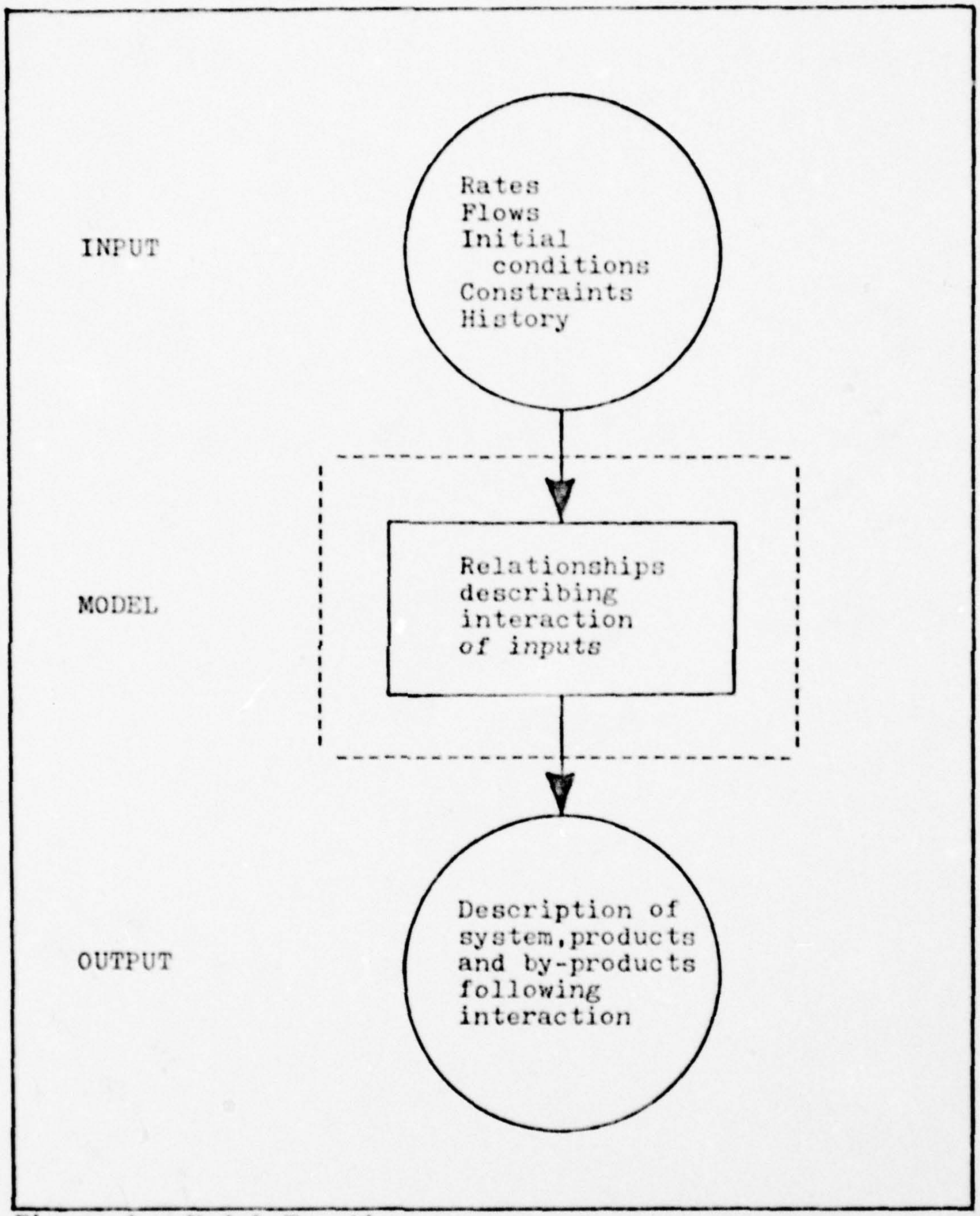


Figure 1. Model Function

TABLE II

Types of Operations Research Problems

Allocation
Assembly Line Balancing
Assignment
Bayesian Decision Models
Big M Method
Binary Linear Programming
Branch and Bound Method
Budget Problems
Classical Optimization
Competitive Model
Convex Programming
CPM
Cutting Plane Method
Decision Theory/Analysis
Deterministic Models
Discrete Programming
Dynamic Programming
Fixed Charge Problems
Game Theory
Geometric Programming
Goal Programming
Graph Theory
Heuristics
Hybrid (deterministic and stochastic)
Implicit Enumeration
Integer Programming
Inventory
Knapsack Problems
Lagrange Multipliers
Linear Optimization
Linear Programming
Markov Chains
Matching Problems
Mixed/Integer Programming
Networks
Nonlinear Optimization
Nonlinear Programming
Parametric Linear Programming
PERT
Plant Location Models
Portfolio Models
Quadratic Programming
Queueing
Random Numbers
Regression Analysis
Reliability
Replacement Models
Routing
Scheduling
Search (for information or decision)
Search Methods (nonlinear)
Sensitivity Analysis
Separable Programming
Sequencing
Set Covering
Simulation
Stochastic Processes
Stochastic Programming
Transportation
Transshipment
Traveling Salesmen
Utility Models/Theory
Zero-One Programming

(Ref 23:67)

Nor can they afford the time. Likewise, managers do not have the ability or time to program computers to perform the work for them. From their perspective, the use of computers and quantitative models is an aid to achieving other objectives, not an objective in itself. In short, a manager's expertise is in management, not in Computer Science. However, the need for managers to be familiar with computers is well established.

An empirical study conducted at the University of Nebraska emphasizes this need (Ref 20). This investigation of the impact of computer generated information on the choice activity of decision-makers suggests that a bias in interpretation exists in the following form. Individuals having little or no experience with computers demonstrate more confidence in information when it is presented in the form of a computer product than when it is not. Alternatively, individuals having a greater degree of experience with computers and computer products demonstrate less confidence in information presented in the form of a computer product. This would indicate that Management students would benefit from a curriculim that increases their confidence in judging the reliability of such information.

Statement of the Problem

The power of quantitative models for decision-making benefit the decision-maker only to the extent that he can

put them to use on matters at hand. The problem is stated in three parts.

First, quantitative models are based on theorems of mathematics (Ref 6:2). Managers are not always able to master and sustain a working proficiency in these subjects. Application of these theorems, then, must be incorporated in the models available to the manager in one of the following ways.

a. The theorems are integrated within the model and applied without the need for direct interaction by the user.

b. The knowledge required for accurate input is condensed to a form consistent with the user's capability to comprehend and apply.

Second, the models often rely on computers to perform complicated and lengthy calculations. Managers must be able to access a computer without extensive education in Computer Science.

Third, managers must be confident that the information generated by quantitative models are useful in the decision-making process. This point deserves added attention where the information is in the form of computer products. Computer generated information should be pertinent and easily interpreted.

Objectives

The goal of this thesis is to demonstrate that properly designed computer programs can give managers access to compli-

cated quantitative models for decision-making. To this end, the following objectives support this goal.

1. Create a computer program for a representative set of quantitative models currently studied by graduate students in the Systems Management program at the Air Force Institute of Technology (AFIT).

2. Develop appropriate instructional material and execution aids for the computer program.

3. Demonstrate that quantitative models for decision-making can be applied by individuals who are unfamiliar with computer operations and the theory underlying these models using interactive programming techniques.

The results of this thesis effort will be presented in the following order:

1. The method used in developing and testing the interactive program and its accompanying instructions.

2. A description of the computer program to include a diagram of its logic, a program listing, and a discussion of its important features.

3. Results of the testing phase to include statistics and comments from the test subjects.

4. A summary and conclusions with recommendations

for future efforts.

It is intended that the results of this thesis will be incorporated into the Graduate Systems Management (GSM) program at AFIT.

II. METHODOLOGY

The Models

As stated in the introductory chapter, this thesis deals with developing a computer program to perform the calculations for quantitative analytical models. Queueing theory provides an excellant source for candidate models. Those selected for this effort are classic queueing theory models which describe service systems characterized by exponential interarrival and service time distributions. Under the classification system of D. G. Kendall, these models are of the M/M/C type (Ref 6:438). Under this classification system, the first M signifies that the model assumes an exponential distribution for the time between customer arrivals (equivalent to a Poisson distribution for customer arrival frequency). The second M signifies that the model assumes an exponential distribution for customer service times. The C, when specified, indicates the number of parallel servers in the system. Six such modeling situations are captured in the computer program. The particular situations deal with different combinations of server number, allowable queue size, and customer population size. The individual models and their characteristics are listed below:

1. Standard M/M/1 Model; describes the steady-state operating characteristics of a service system with one server. No limits are placed on the size of

the queue or the customer population.

2. Finite Queue M/M/1 Model; describes the steady-state operating characteristics of the Standard M/M/1 Model in which the queue is restricted to a finite length.

3. Finite Population M/M/1 Model; describes the steady-state operating characteristics of the Standard M/M/1 Model in which the calling population is of a finite size.

4. Standard M/M/C Model; describes the steady-state operating characteristics of a service system with two or more identical servers. No limits are placed on the size of the queue or the customer population.

5. Finite Queue M/M/C Model; describes the steady-state operating characteristics of the Standard M/M/C Model in which the queue is limited to a finite length.

6. Finite Population M/M/C Model; describes the steady-state operating characteristics of the Standard M/M/C Model in which the calling population is of a finite size.

Two logical extensions to these models exist. The first of these applies to single server systems in which the time needed to service a customer can be controlled or adjusted. Given that there is a cost associated with having customer

units wait in the queue in anticipation of service as well as a marginal cost associated with providing that service; it can be demonstrated that there exists a particular rate of service which yields the lowest overall cost per unit time for operating the system. This rate, the optimal service rate, is a function of the customer arrival rate, the cost of having customer units wait and the marginal cost of servicing each customer unit. This marginal cost of service is the same as the variable cost of providing service to each unit as distinquished from the fixed cost of making the service available.

The second extension applies to multiple server systems in which there exists an option to vary the number of servers provided. Again, the object is to minimize the overall cost of operating the service system; this time by selecting the optimal number of servers based on the cost of having units waiting in the queue, the expected number of units in the system and the cost of providing the individual servers.

A complete discussion of these models and the classification system can be found in most undergraduate and graduate level texts on Operations Research or Queueing Theory. The program developed in this thesis uses the equations presented by Budnick, Mojena, and Vollman in their text, <u>Principals of Operations Research for Management</u> (Ref 6:429-474).

Programming Objectives

The program itself is designed for use at a time sharing terminal. This allows the program to assist the user with inputting the necessary data for model selection and program execution. The intent is to lead the user through a sequence of questions and commands which will result in an immediate solution to the desired model parameters. The inputs required for program execution are number of servers, allowable queue length, population size, customer arrival rate, and service rate. The objective, that the program be easy to use, requires that the input format not be overly restrictive. That is to say, a background in computer language should not be a prerequisite for use. Similarly, the resulting output must be easily understood.

Operating Instructions and Aids

Detailed operating instructions for the program are included in Appendix C of this thesis. They include:

1. Step by step instructions for accessing the computer facilities at the AFIT School of Engineering, using the time sharing remote terminals.
2. A brief description of the program and the types of problems it solves.
3. An explanation of the inputs required for program execution.

4. An explanation of the outputs generated by the program.

5. Examples of all phases of program operation.

Testing of Objectives

The program and its accompanying instructions were tested to ascertain the extent to which they meet stated objectives. Five subjects of varying education levels and computer experience were asked to use the program to solve representative queueing theory problems. The program and instructions are evaluated on the basis of the ability of each subject to operate a remote terminal, input information correctly, and interpret the output; as well as the confidence they place in the resulting information. The subjects were placed in a room with a remote terminal and asked to complete two situational problems involving service systems. These problems are listed in Appendix C. The only aids allowed were the written instructions for the program contained in Appendix C. After the test session, the subjects provided information by completing the questionaire shown in Appendix D. The results of this test are reported in Chapter IV. A detailed explanation of the program is provided in the next chapter.

III. PROGRAM DESCRIPTION

Program Elements

The specific program developed for this thesis effort is written in FORTRAN IV. It is divided into nine elements as follows:

1. The primary program, MAIN, directs all user interactions to include; model selection, problem inputs and solution printouts.
2. A subroutine, WRONG, which aids the user in correcting erroneous responses to "yes" or "no" questions.
3. A function, FACT, which computes factorials used by other program elements.
4. A subroutine, DATA1, which compares observations of customer arrivals with the distribution assumptions of the models.
5. A subroutine, DATA2, which compares observations of service times with the distribution assumptions of the models.
6. A subroutine, PROBS, called by DATA1 to compute Poisson probabilities.
7. A subroutine, PROBE, called by DATA2 to compute exponential probabilities.
8. Six subroutines corresponding to the six discrete models referred to in Chapter II:

MODEL	SUBROUTINE
Standard M/M/1	STMM1
Finite Queue M/M/1	FQMM1
Finite Population M/M/1	FPMM1
Standard M/M/C	STMMC
Finite Queue M/M/C	FQMMC
Finite Population M/M/C	FPMMC

9. A subroutine, OPTMU, which computes the optimal service rate for all service systems restricted to one server (M/M/1 type models).

10. A subroutine, OPTC, which computes the optimal number of servers for multiple server systems (M/M/C type models).

All numerical inputs to the program are free of format restrictions. That is to say, decimal fractions may be entered in standard form and the use of decimal points with whole numbers is optional. Due to the nature of the modeling theory, negative values are meaningless as inputs and are never entered.

The following sections describe the logic flow used in the program. Reference to Appendix A, Definitions of Variables, will aid in comprehending these discussions. Figure 2 presents a simplified logic diagram of MAIN. A more detailed logic for all elements is depicted in the flow diagrams in Appendix B. Examples of executed programs are contained in Appendix C.

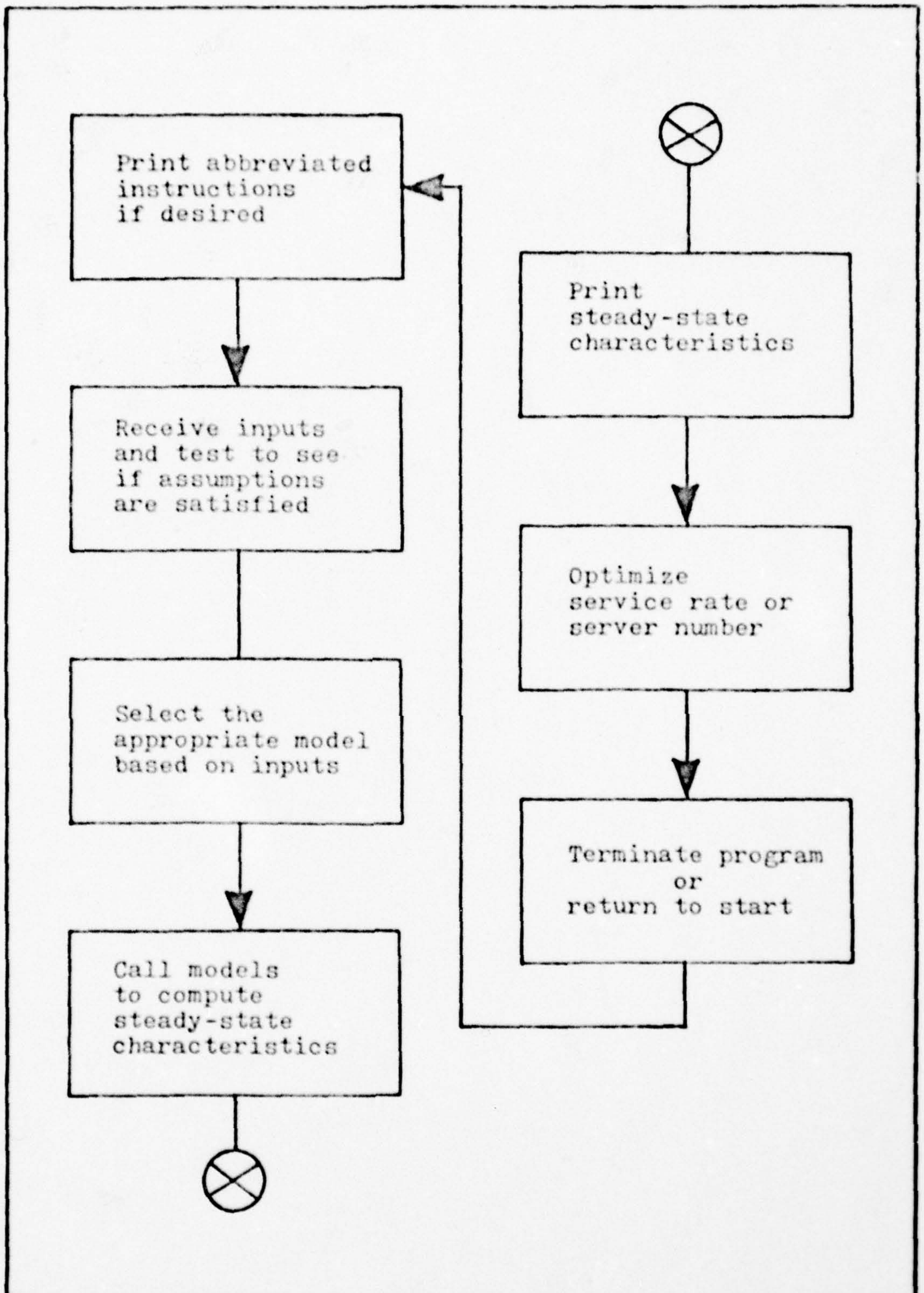


Figure 2. Simplified Program Logic

Program MAIN

The basic operations of MAIN are:

1. Select the appropriate model.
2. Receive inputs describing the specific service system.
3. Direct the appropriate subroutine to compute steady-state characteristics for the service system.
4. Echo the input data for verification and print the steady-state characteristics.
5. At the option of the user, call a subroutine to optimize service rate or server number as appropriate.
6. At the option of the user, run additional problem sequences or terminate the program.

At the program beginning, the six models are listed for the user along with the identifying code numbers. At this point, the user is given the option of selecting a desired model by transmitting its corresponding code. Alternatively, if the user is not certain which model best fits his particular service system, a sequence of program inquiry will select the model for him. The first input for this sequence is server number (C). An input of one server results in the program variable MSIGN taking a value of 0. An input of two or more servers results in MSIGN taking the value of 3. Effectively, this divides the

models into two groups, the M/M/1 type and the M/M/C type, respectively. The user is then asked if the queue is limited to a finite number. If so, a program variable MVAL is given the value 2. If the queue is not limited, then possibly the "customer" population is limited to a finite number. If this is the case, MVAL is given the value 3. If neither the queue nor the customer population is limited, MVAL equals 1 by default. MSIGN and MVAL are then added together and the sum corresponds to the correct code for the best model. From this point on MVAL identifies the chosen modeling situation.

Computation of steady-state characteristics by the six subroutines requires the input of the following parameters:

PARAMETER	VARIABLE NAME
1. Server number	C
2. Maximum queue length (finite queue models)	LIMQ
3. Customer population size (finite population models)	M
4. Customer arrival rate	LAMBDA
5. Service rate	MU

C, LIMQ, and M are entered during model selection.

LAMBDA and MU may be entered directly into the program, or sample data concerning arrivals and service times may be entered instead. When sample data is entered, it is checked to see if the arrivals and service times are distributed in accordance with the model assumptions and

LAMBDA and MU are computed from that data. These checks and computations are performed by the subroutines DATA1 and DATA2 which are discussed later. It should be noted that, in case of the finite population models, the input LAMBDA refers to the mean arrival rate of each individual population member. In the remaining models, LAMBDA measures the mean arrival rate of all customers to the service system.

Once the inputs are transmitted, the program calls the appropriate model subroutine to compute the steady-state characteristics which are then returned to MAIN to be included in the output listing. The logic flow for these subroutines is discussed later in this chapter.

The output listing begins with an echo of the input parameters. This allows the user to verify that the system is being described as intended and also provides a complete model description separated from the record of user/program interaction. The steady-state characteristics included in the listing vary according to the model used. For example, the probability of customer rejection due to a full queue (PNMAX) applies only to finite queue systems and is included accordingly. When describing Standard M/M/1 and Standard M/M/C systems, it is possible that a steady-state will not be reached. This occurs when the customer arrival rate exceeds the ability of the service system to handle the arrivals. Specifically, this happens whenever

LAMBDA C x MU (1)

In such cases, the output states this explicitly.

After the output listing, the user may elect to call for optimization of MU for a single server system or optimization of server number (C) for a multiple server system, as discussed in Chapter II. The specific logic of this process is discussed later in this chapter.

At the end of the optimization sequence, the user may choose to run another problem. If he does not so choose, the program terminates.

Subroutine WRONG

WRONG is called by MAIN whenever it detects an unacceptable response to a "yes" or "no" question. MAIN reads only the first character in the response to such a question. If the first character is anything but "Y" (yes) or "N" (no), then, WRONG is called. WRONG transmits an error message to the user and asks for a corrected response which is then evaluated for acceptability. If the corrected response is "Y" or "N", it is returned to MAIN and the program continues. If not, the error sequence is repeated until the user transmits an acceptable response.

Function FACT(K)

The characteristic equations for the six queueing models use factorial computations in their solution. Where

appropriate, the model subroutines call FACT(K) to compute the factorial of a given integer. When computing the factorial of K, the function first initializes the variable FACT and then uses a DO loop to iteratively multiply FACT by successive integers from 1 to K. The result is then returned to the calling subprogram.

<u>Model Subroutines</u>

The six model subroutines apply the equations derived from the queueing theory. They are, necessarily, as complex as the algebraic computations which they perform. A comprehensive understanding of these subroutines requires reference to the flow diagrams in Appendix B while examining the individual FORTRAN statements one by one. Input and output variables are transmitted between MAIN and the subroutines by the use of both arguments to the subroutines and COMMON statements. Variable names are consistent throughout the entire program. Where appropriate, program variables are chosen to reflect the quantity they represent. For instance, NMAX is the maximum number of units allowed in a finite queue system. The subprograms accurately reproduce the equations presented in the reference text (Ref 6:441, 471-474). In addition to MAIN, the subprogram OPTC also calls the M/M/C models.

<u>Subprogram OPTMU</u>

OPTMU is a self sufficient subprogram. It requests

input for the cost of waiting (CWAIT) and the marginal cost of service (CSERVE), and then computes and prints out the optimally cost effective value of MU through the relation-ship

$$MU = LAMBDA + (LAMBDA \times CWAIT/CSERVE)^{\frac{1}{2}} \qquad (2)$$
(Ref 6:460)

where CWAIT is defined as the cost per customer per unit time associated with having a customer unit in the service system and CSERVE as the marginal cost of providing service to each customer.

Subprogram OPTC

OPTC uses CSERVE and CWAIT to compute the most cost effective number of servers for a multiple server system. It uses the following expected total cost per unit time function:

$$ZEE = CSERVE \times C + CWAIT \times ELESS \qquad (3)$$
(Ref 6:461)

where ELESS is the steady-state mean number of units in the system. It should be noted that CSERVE here is redefined as the cost of providing each server per unit time. The subprogram operates in an iterative fashion where CSERVE and CWAIT remain constant and ELESS varies with iterative values of C. The process works in the following manner. C is assigned the value of 2. The appropriate model

subprogram is called to determine the value of ELESS corresponding to C = 2. ZEE is then computed. C is then set to 3. ELESS and ZEE are again computed. If ZEE corresponding to C = 2 is less than or equal to ZEE corresponding to C = 3, then C = 2 is taken as the optimal value (C which results in minimum cost). If not, the process is repeated for C = 4, C = 5, etc. until a ZEE is found which is greater than the preceeding ZEE. The optimal number of servers is the next to last C. This iterative process quarantees that the optimal C chosen poduces a cost (ZEE) less than the cost for values of C both preceeding and following it. The optimal C and its associated cost are reported to the user and program control is returned to MAIN. Note that before the iterative steps are initiated, values of C are examined and the process begins with the first C that allows the system to reach steady-state. Recall that arrival rate must be less than combined service rate for the Standard M/M/C model.

Subroutine DATA1

DATA1 operates on raw observation data concerning customer arrivals (for finite population models, see DATA2 below). At the user's discretion, DATA1 accepts the number of customer arrivals observed during any number of equal observation periods, up to 100 periods. This data is used to compute the mean arrival rate (LAMBDA). This data is compared to a Poisson probability mass function, using

a one tail chi-square goodness-of-fit test with a .01 level of significance. Three possible outcomes result from the test:

1. Acceptance of the null hypothesis that the data fits the Poisson distribution.
2. Rejection of the null hypothesis.
3. The conclusion that an insufficient number of observations were made for the test to be valid.

In the event of 2 and 3 of the above, the user has the option of continuing the problem based on the computed LAMBDA. If sample data is not entered, the program accepts a value of LAMBDA by direct input.

Subroutine DATA2

DATA2 operates on raw data concerning service times. At the user's descretion, DATA2 accepts the time needed to service up to 100 customers. A mean service time (MU) is computed. Service time data is compared to an exponential probability function in the same manner as the customer arrival data. Possible results and options are the same as in the DATA1 subroutine. In the case of customer arrivals from a finite population, the interarrival times for individual customer units are entered and tested in the same method as service times. A mean arrival rate per customer is calculated for use in the FPMM1 and FPMMC subroutines.

A necessary complement to the program described in this

chapter is a set of instructions for using the program. Appendix C contains a copy of these instructions. The program and instructions together enable users to apply the models to problem situations. An evaluation of the final product appears in the following chapter.

IV. RESULTS OF TESTING

The sample problems located in Appendix C were used in the program test. A review of those problems and their accompanying explanations will help in understanding protions of the following analysis.

Test Subjects

Five test subjects evaluated the program described in Chapter III. These subjects reported pertinent background information concerning education, vocation, computer experience and knowledge of queueing theory on the questionnaire located in Appendix D. This information is summarized in Table III. The subjects represent a wide range of vocational and educational experience. Computer experience for subjects A and E is limited to basic programming courses. Subjects C and D reported extensive computer experience, however, this experience is limited to data storage and retrieval systems. Data analysis is not part of their job-related computer experience. Their jobs do require them to use remote terminals similar to the one used for the test sessions. None of the subjects reported any significant knowledge of queueing theory.

TABLE III.

Background Data For Test Subjects

	A	B	C	D	E
Vocation	receptionist	secretary	data processor	Engineer	high school student
Education	2 years undergraduate	high school graduate	2 years undergraduate	MBA	grade 11
Knowledge of Queueing Theory	none	none	limited	limited	none
Computer Experience	limited	none	extensive	extensive	limited
Remote Terminal Experience	limited	none	extensive	moderate	limited

Individual Task Performance

Table IV.summarizes the extent of difficulty each subject experienced while accomplishing individual tasks during the test sessions.

TABLE IV.

Summary of Difficulties Encountered During Program Test

Task	None	Slight	Moderate	Extreme
Controls/ Keyboard	A,B,C,E,	D		
LOGIN	C,D,E	A,B		
Program Access	A,C,D,E	B		
Data Input	A,C	B,D,E		
Model Selection	A,B,C,E		D	
LOGOUT	A,B,C,D,E			
Interpretation of Results	A,C,E	B,D		
Printed Instructions	A	B,C,E	D	

Controls/Keyboard: None of the subjects experienced any significant difficulties operating the terminal controls. Subject D could not locate the carriage control key even though it is clearly marked "RETURN" and its orange color contrasts with all the other keys.

LOGIN Procedure: Problems with this task were minimal. Subject A did not place commas between the parameters. Subject B used so much time to make the input that the DATA PHONE connection was automatically terminated for inactivity. The second attempts for both subjects were successful.

Program Access: Only one problem ocurred during program access. Subject B typed the commands incorrectly twice. The resulting error messages generated confusion.

Data Input: Subjects A and C carefully studied each request for input. Their responses were correct, although they expressed concern over being unsure about the definitions of arrival rate and service rate while working Problem 1. The other subjects expressed the same confusion and input service rate data in place of arrival data. Their second attempts were successful.

Model Selection: Subjects A,B,C, and E used the program sequence in which a series of questions leads the user to the correct model. These four subjects followed the sequence correctly for both problems. For Problem 1, Subject D chose to select the model himself. He selected a finite queue model reasoning that if there were only 25 drivers in the population, then the queue could be no longer than 25. For Problem 2, the program led him to the correct model.

LOGOUT: The LOGOUT procedure presented no problems.

Interpretation of Output: Both sample problems ask a specific question. In each case, the test subjects surmised the correct answer based on the output they received. However, all subjects indicated they were unsure of the meaning of certain printed steady-state characteristics. The problem is discussed further in the following section.

Printed Instructions: Subjects A,B, and C found that the printed instructions were adequate. Subjects D and E commented on the survey that the instructions did not adequately define the terms used in the program. Two observations are worthy of note. First, subjects D and E spent the least amount of time reading the instructions before executing the program. Second, even though subjects A,B, and C rated the instructions favorably on the survey, their verbal comments after the test sessions indicated that they had the same problems with definitions as reported by subjects D and E.

Analysis of Results

Overall, operation of the terminal, to include accessing the program, was satisfactory. Some mistakes were made initially, but the subjects mastered the basic terminal operating procedures with a minimum of effort. Those that followed the instructions carefully and deliberately were successful on the first attempt. Subject D provided the contrast. Rash actions caused him to make several mistakes.

For the other subjects the step-by-step process of gathering inputs successfully led the subjects through the operation of the program. That is to say, the subjects were able to initiate the program and follow it through to a logical conclusion. However, not all subjects arrived at the correct solutions. There are two significant reasons for this.

First is a problem of definition. The printed instructions and the instructions within the program were not sufficiently clear to guarantee that all subjects understood exactly what was asked of them. None of the subjects were absolutely certain that they understood the difference between finite queue systems and finite population systems. Subject D arrived at an incorrect solution to Problem 1 due to this confusion.

The second reason involves interpretation. Errors made on Problem 2 provide the best illustration. Subject B identified the system as having four servers (the four telephone lines) instead of one server (Joe). Subject E failed to convert the stated fact that Joe averaged 30 seconds per customer into a service rate of 2 customers per minute. Finally, Subject D was unable to recognize the fact that one phone line would be used to service a customer thus limiting the maximum length of the finite queue to three. His input was a maximum queue length of four.

One final test result should be considered. All five

test subjects indicated that they felt a high degree of confidence in the value of the program to aid decision-making. On a scale of 1 (least confident) to 5 (most confident) every subject chose a confidence rating of 4.

Conclusions drawn from these test results are reported in the next chapter.

V. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

Modern management theory recognizes the need for quantitative decision-making in routine management efforts. Limited resources often preclude managers from applying quantitative techniques to their routine problem solving situations. They do not possess the requisite technical knowledge for sophisticated decision analysis, nor do they have access to those who do. Rapid advances in computer technology and quantitative problem analysis create a need for developing better methods of application at the practical level. One such method invloves the use of interactive computer programming. This research explores the possibility of using interactive programming to satisfy this need.

An interactive program was written to integrate complex queueing theory models into a format that is accessible to non-technically oriented managers. The program was tested to see if the following objectives were satisfied.

1. Proper application of mathematical theory.
2. The user need not have an extensive knowledge of computer operations.
3. Program inputs and outputs have a meaningful interpretation for the non-technical user.
4. The user feels confident that the results are valid.

Analysis of the test results lead to the following conclusions.

Conclusions

Three major conclusions emerge from this research. They are:

1. Interactive programs can present complex analysis models in a format understood by a non-technical user. An extensive background in the underlying mathematical theory from which the models are developed is not a prerequisite for the user to apply the program to situational problems.
2. The manner in which the program communicates the model concepts and sequential instructions to the user are of paramount importance in developing an analysis sequence that is meaningful to the decision-maker. This means that all user-program interactions must be consistent with the user's experience. Even though terms and concepts that are new to the user are well defined within the program, they represent possible areas of confusion, and therefore, tend to limit the efficiency with which the program is employed. The language of the program should be the language of the user. Inputs, outputs, and intervening actions must be considered in this context.

3. An extensive knowledge of computer operations is not required to use interactive programs. A decision-maker can be involved directly with the analysis process.

Recommendations

The product of this research serves only to indicate that the benefits of quantitative analysis can be made available to decision-makers through interactive programming. There are many logical extensions of this concept. Some of them are listed below.

1. Survey Air Force activities to classify potential users. Such a survey might identify those specific analytical models with the greatest potential for application.
2. Identify the characteristics of an interactive program that contribute most to its effective operation. This could lead to the development of interactive program design principles.
3. Develop similar programs for use in the AFIT Systems Management curriculum. Exercises in practical application could be generated for courses in management, economics and accounting.
4. Although far from conclusive, the results of this research suggest that direct interaction with the analysis program influences the amount of confi-

dence a user displays in the resulting information. Design and conduct a research experiment to determine the extent to which interaction affects that confidence.

Bibliography

1. Ackoff, Russell L. and Patrick Rivett. A Manager's Guide to Operations Research. New York: John Wiley and Sons, 1963.

2. Ackoff, Russell L. and Maurice W. Sasiene. Fundamentals of Operations Research. New York: John Wiley and Sons, Inc., 1968.

3. American Management Associations. Directory of Management Education Programs. New York: American Management Associations, 1977.

4. ASD Computer Center INTERCOM GUIDE (Revision A). Wright-Patterson Air Force Base, Ohio: United States Air Force, August 1976.

5. Boyes, Rodney L., Robert W. Shields, and Larry G. Greenwell. Introduction to Electronic Computing: A Management Approach. New York: John Wiley and Sons, Inc., 1971.

6. Budnick, Frank S., Richard Mojena, and Thomas E. Vollman. Principals of Operations Research for Management. Homewood, Illinois: Richard D. Irwin, Inc., 1977.

7. Carter, L. R. and E. Huzan. A Practical Approach to Computer Simulation in Business. New York: John Wiley and Sons, Inc., 1973.

8. Chase, Richard B. and Nicholas J. Aquilano. Production and Operations Management. Homewood, Illinois: Richard D. Irwin, Inc., 1977.

9. Digital Computer Manual for Faculty and Students of the School of Engineering (4th Edition). Wright-Patterson Air Force Base, Ohio: United States Air Force, Air University, September 1978.

10. Drucker, Peter F. The Age of Discontinuity. New York: Harper and Row, 1978.

11. George, Claude S., Jr. The History of Management Thought. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1968.

12. Giffin, Walter C. Queueing. Columbus, Ohio: Grid, Inc., 1978.

13. Gregory, Carl E. The Management of Intelligence. New York: McGraw-Hill Book Company, Inc., 1967.

14. Hein, Leonard W., PH.D., C.P.A. The Quantitative Approach to Managerial Decisions. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1967.

15. Herrmann, Cyril C. and John F. Magee. "Operations Research for Management," Readings in Management, Fourth Edition, 178-193. Cincinatti: South-Western Publishing Company, 1974.

16. Huysmans, Jan H. B. M. The Implementation of Operations Research. New York: Wiley-Interscience, 1970.

17. Johanson, Robert, Jacques Vallee, and Kathleen Spangler. Electronic Meetings: Technical Alternatives and Social Choices. Reading, Massachusetts: Addison-Wesley Publishing Company, 1979.

18. Laird, Dugan and Richard C. Grote. Solving Managerial Problems. Reading, Massachusetts: Addison-Wesley Publishing Company, 1971.

19. Lindsay, Franklin A. New Techniques for Management Decision Making. New York: McGraw-Hill Book Company, Inc., 1958.

20. Luthans, Fred and Robert Koester. "The Impact of Computer Generated Information on the Choice Activity of Decision Makers," Academy of Management Journal, 19,2: 328-332 (June 1976).

21. McMillan, Claude, Jr. Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machines. New York: John Wiley and Sons, Inc., 1970.

22. Moan, Floyd E. "Does Management Practice Lag Behind Theory in the Computer Environment?" Academy of Management Journal Sciences, 16: 7-23 (March 1973).

23. Schumacher, Robert M. "Survey and Extensions of Computer Resources to Support Graduate Operations Research Education," Thesis. Wright-Patterson Air Force Base: Air Force Institute of Technology, 1978. AFIT/GOR/SM/78D-13.

24. Terry, George R. Principals of Management. Homewood, Illinois: Richard D. Irwin, Inc., 1977.

25. Thesen, Arne. Computer Methods in Operations Research. New York: Academic Press, 1978.

APPENDIX A

DEFINITIONS OF VARIABLES

PROGRAM VARIABLE	MODEL PARAMETER	DEFINITION
A	LAMBDA	Mean arrival rate (units per time period)
ARRIV	-	A 100 element array containing customer arrival information
ASUM	-	Program storage equal to the sum of ARRIV elements (total observed arrivals)
B	MU	Mean service rate (units per time period)
C	C	Number of servers
CHI2	-	Program storage used to accumulate the chi-square test statistic for sample data
CHISQ	-	A 37 element array containing discrete values of the chi-square distribution
CPRIMS	C'S	Cost per server per unit
CRHO	-	Program storage equivalent to RHO/C
CSERVE	CS	Marginal cost of service per customer
CSIZE	-	Program storage used to compute cell size for the chi-square test
CSUM	-	Program storage used to compute iterative totals of CSIZE
CWAIT	CW	Cost of waiting per customer-unit time

DPZERO	-	Program storage
ELBEE	LB	Steady-state mean number of units in queue for busy system
ELESS	LS	Steady-state mean number of units in system
ELQUE	LQ	Steady-state mean number of units in queue
EVAL	-	Program storage used to compute the expected value of customer arrivals or service time based on probability functions
FREQ	-	A 100 element array containing the frequency of a value of ARRIV or STIM
I	-	Internal program index
INT	-	Internal program index
J	-	Internal program index
K	-	Internal program index
L	-	Internal program index
LEFT	-	Internal program index
LIM1	-	Internal program index defining all boundaries
LIM2	-	Internal program index defining cell boundaries
LIMQ	-	Maximum number of units allowed in a finite queue
M	M	Number of units in a finite population
MSIGN	-	Program storage used to define model codes
MVAL	-	Program storage used to define model codes

NI	-	Internal program index equal to the number of data observations
NCELL	-	Internal program index equal to the number of cells used in the chi-square test
NDF	-	Number of degrees-of-freedom for the chi-square test
NMAX	N	Maximum number of units allowed in system equal to LIMQ + C
NMC	-	Program storage equal to NMAX - C
OPTVAL	(MU)	Optimal MU for single server systems
PBUSY	-	Program storage equal to probability that the system is busy (1 - PZERO)
PNGEC	-	Program storage equal to the probability that all servers are busy
PNMAX	P(N)	Probability of rejection from the system due to a full queue (finite queue systems)
POISS	-	Storage for discrete Poisson distribution values
PROB	-	Probability value
PROB1	-	Probability value
PROB2	-	Probability value
PSUM	-	Internal program storage summing probability values.
PZERO	P(0)	Probability that the system is idle

STIM	-	A 100 element array containing observed service times
TLIM1	-	Internal program index defining cell boundaries
TLIM2	-	Internal program index defining cell boundaries
TSUM	-	Program storage equal to the sum of STIM elements (total observed service time)
Q	-	Program storage-answers to yes/no questions
RHO	RHO	Traffic intensity equal to LAMBDA/MU
W	-	Program constant equal to 0
X	-	Program constant equal to 0.5
Y	-	Program constant equal to 1.0
Z	-	Program constant equal to 2.0

APPENDIX B

LOGIC DIAGRAMS AND PROGRAM LISTING

The following list is provided for the convenience of the reader.

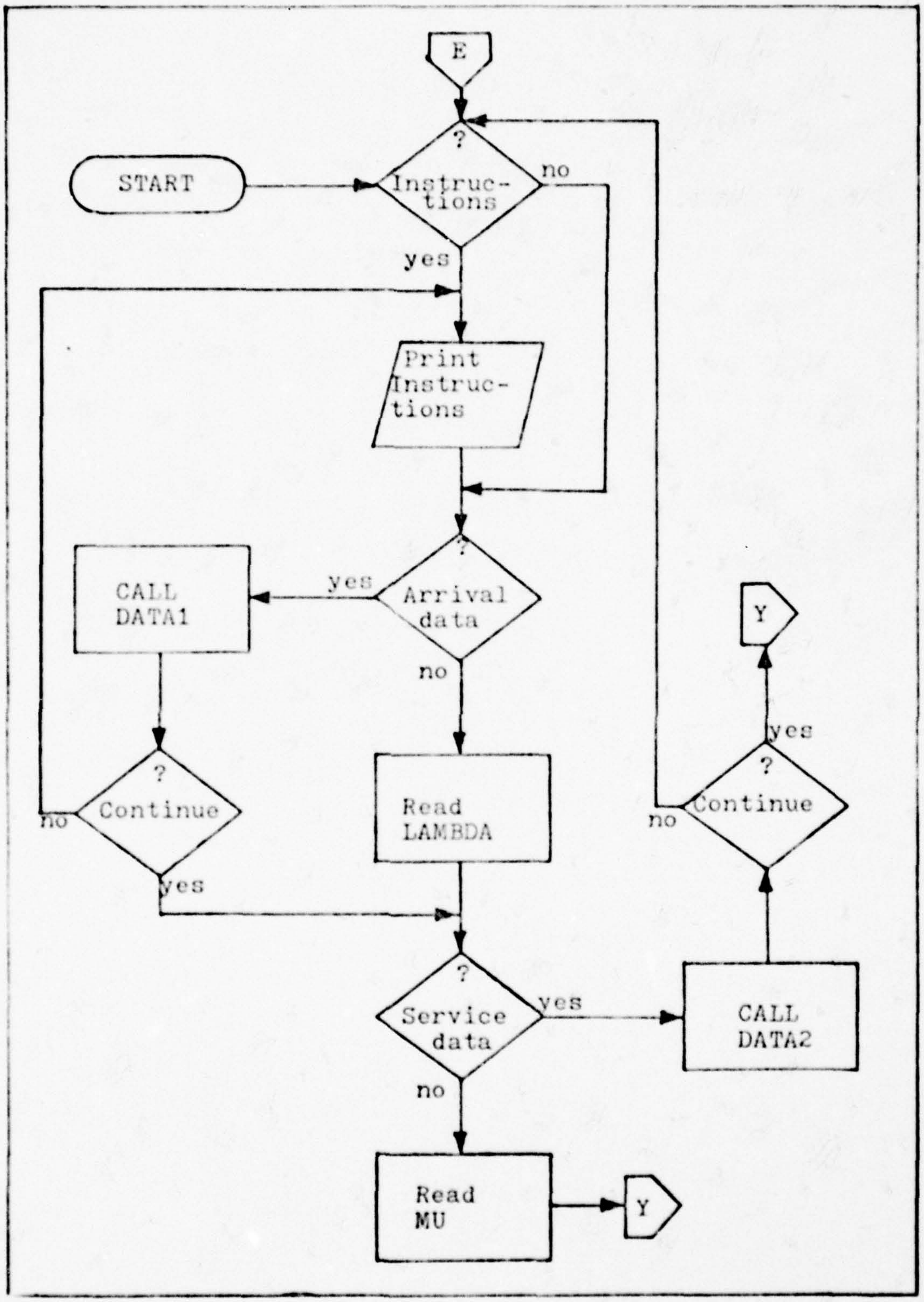


Figure 3. Program MAIN

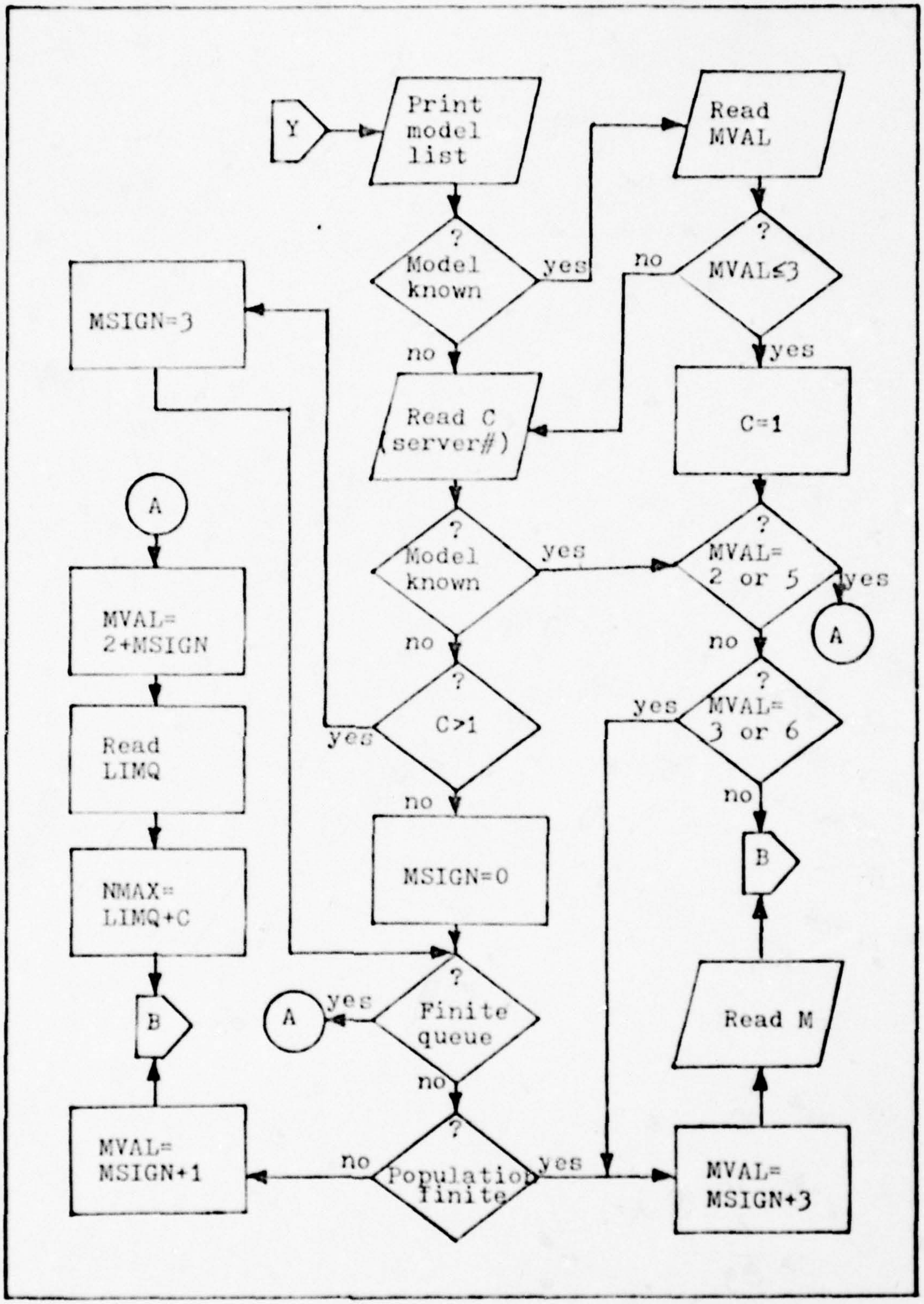


Figure 3. Program MAIN (cont.)

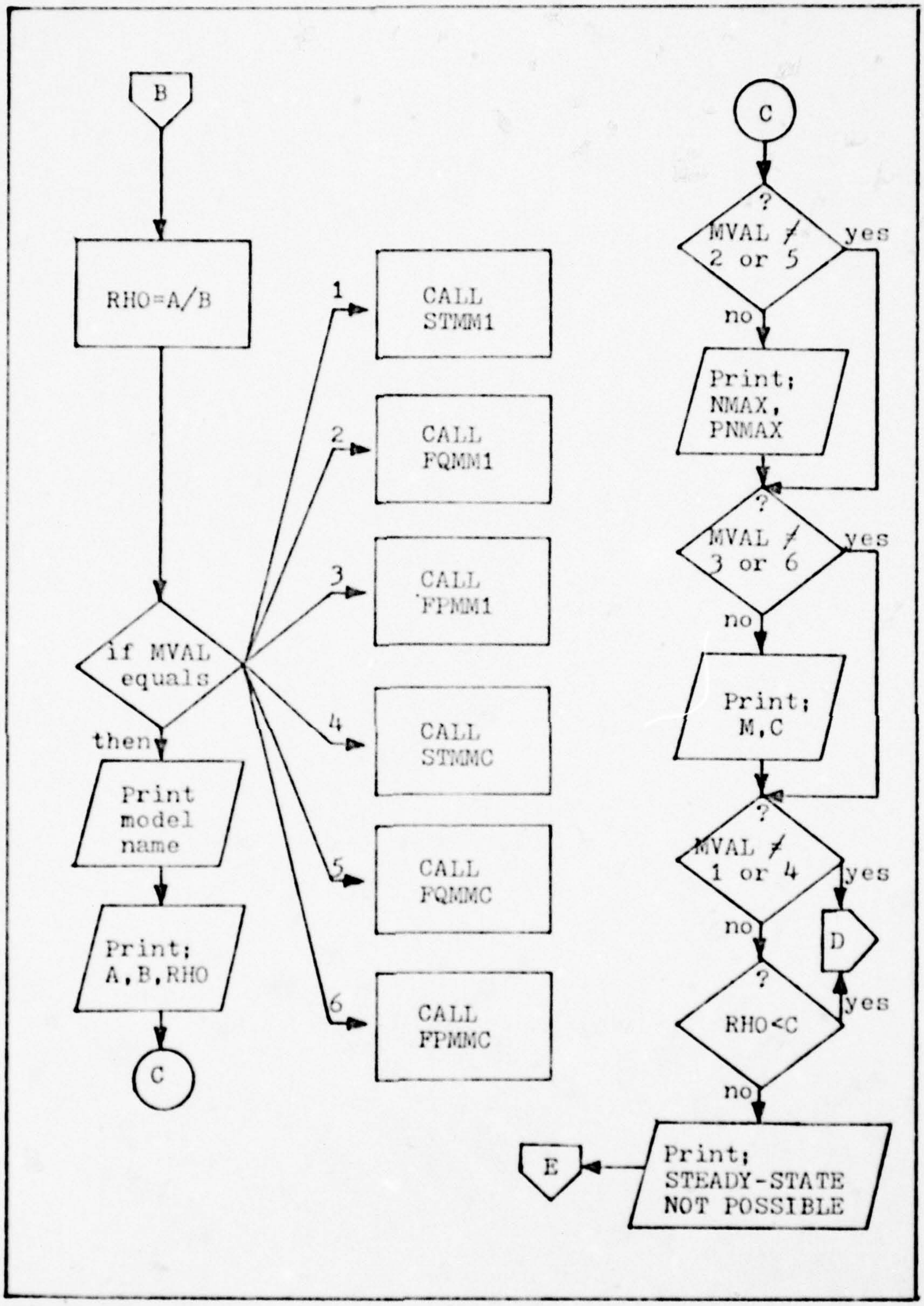


Figure 3. Program MAIN (cont.)

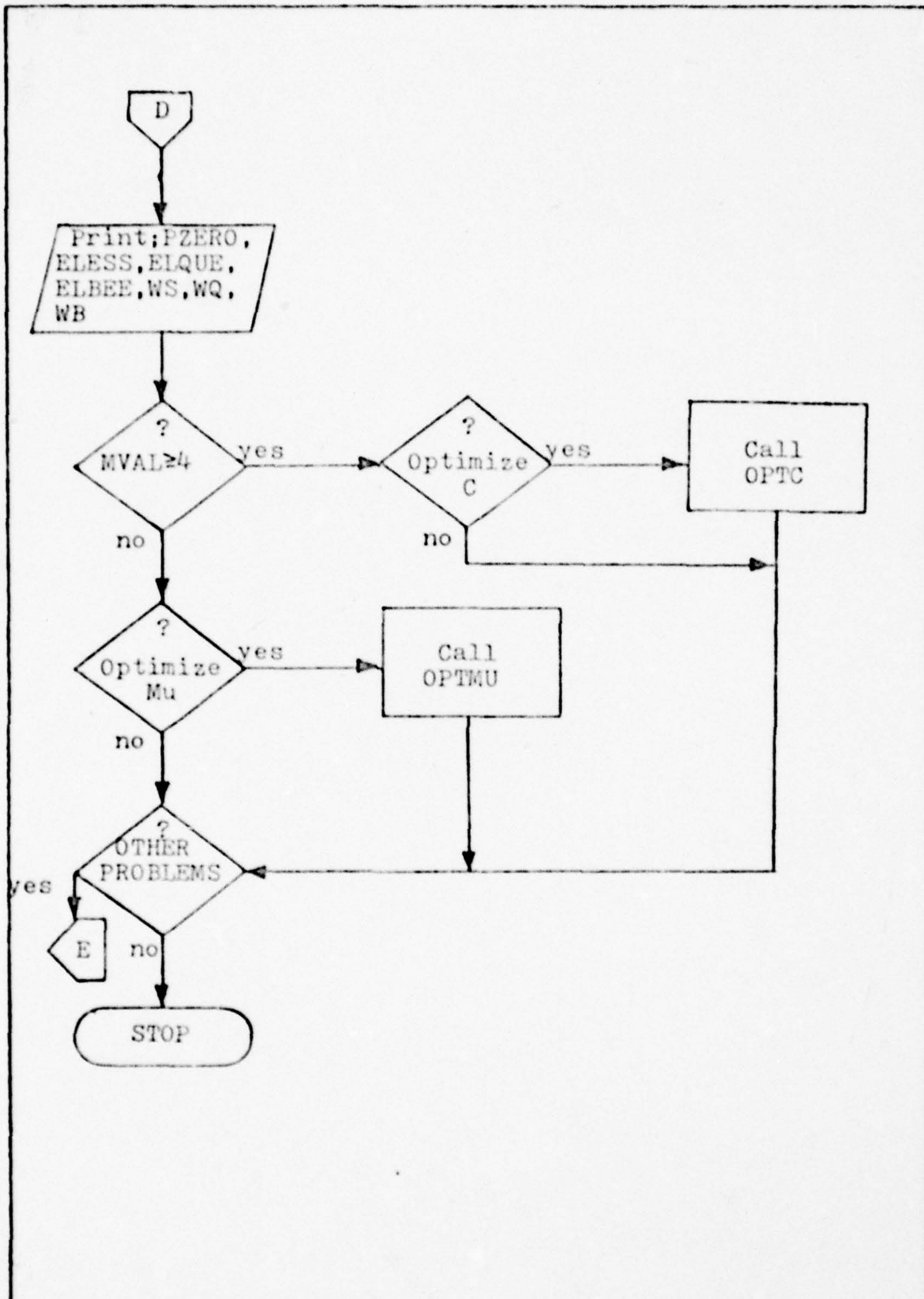


Figure 3. Program MAIN (cont.)

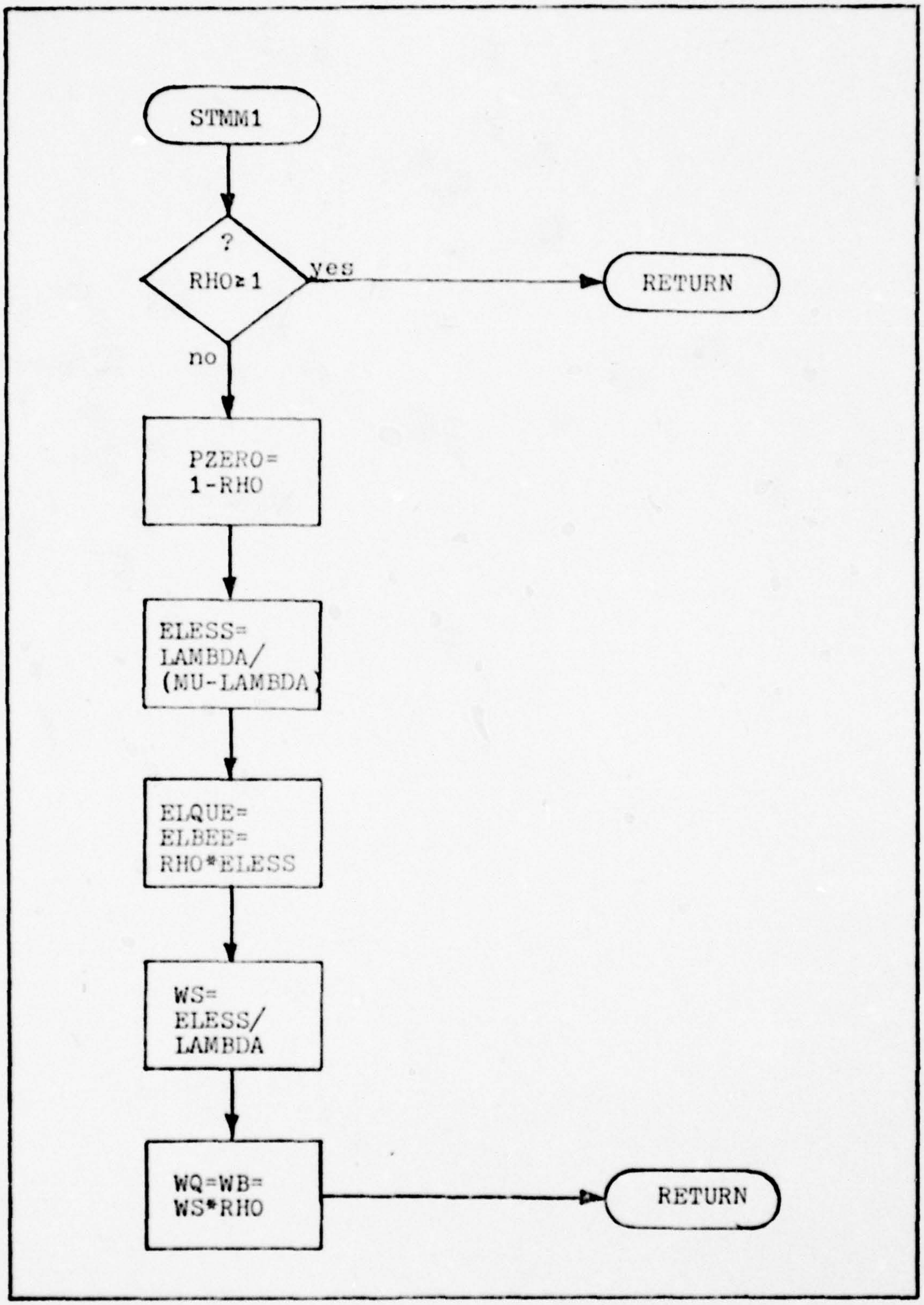


Figure 4. Subroutine STMM1

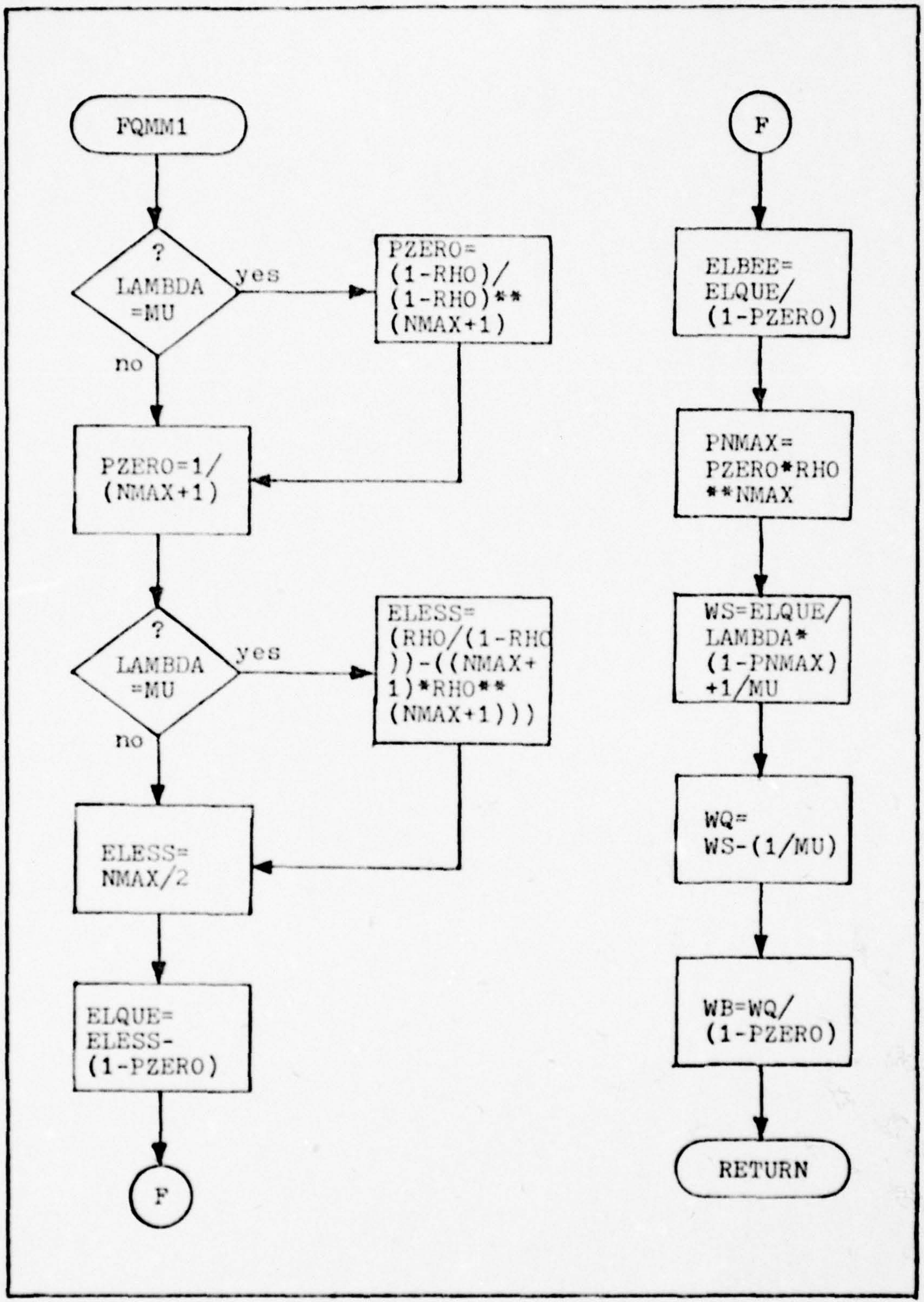


Figure 5. Subroutine FQMM1

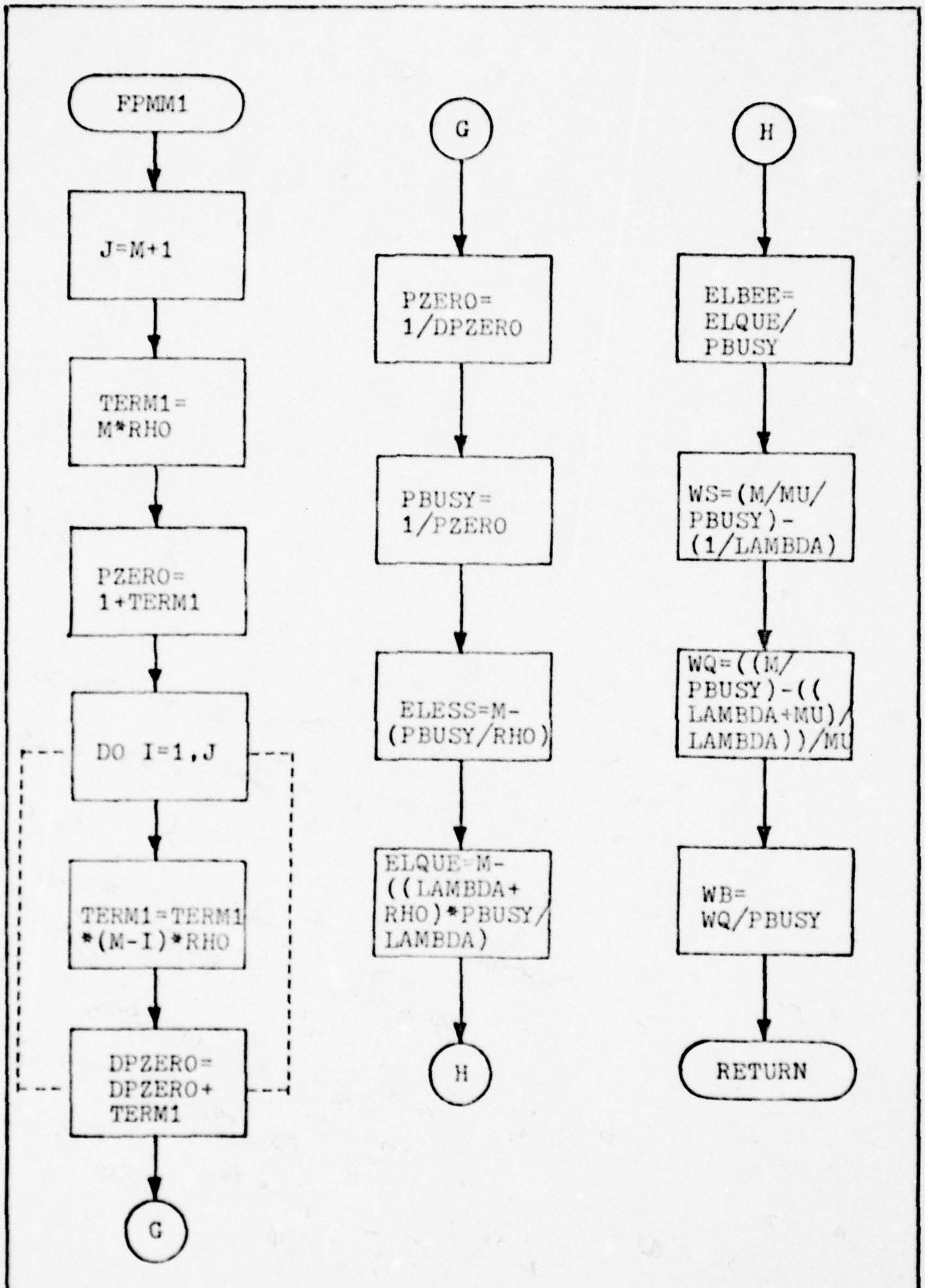


Figure 6. Subroutine FPMM1

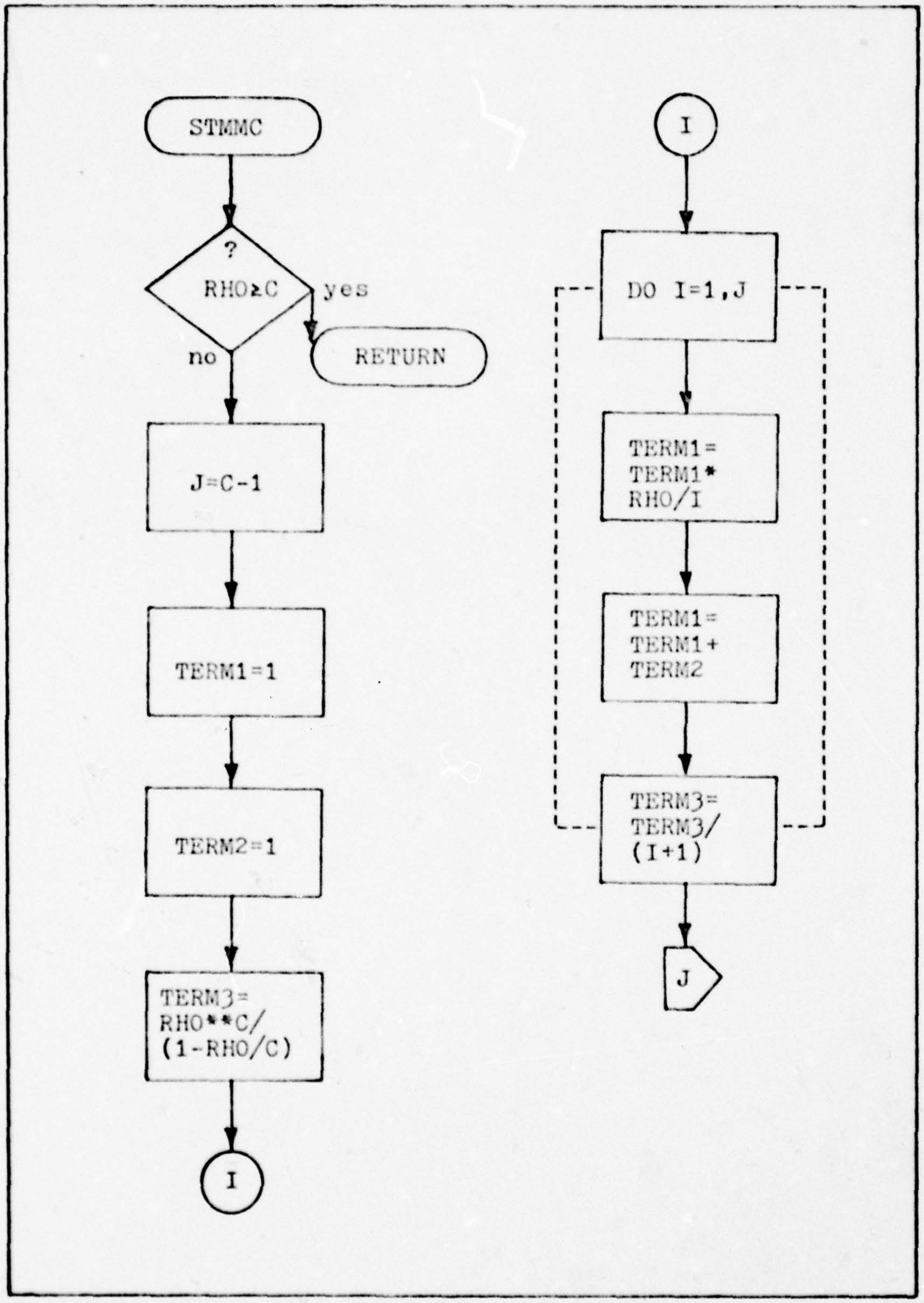


Figure 7. Subroutine STMMC

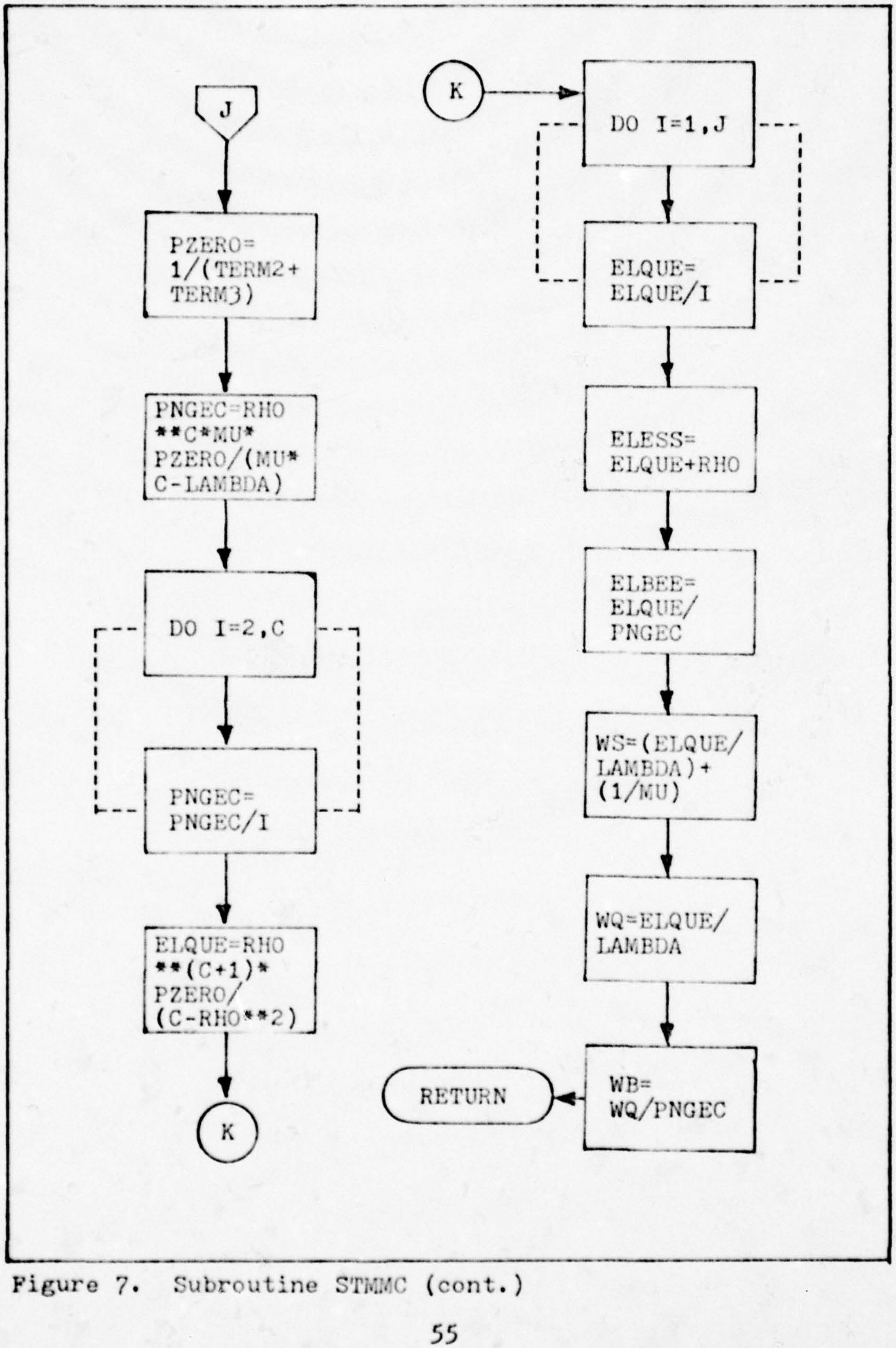


Figure 7. Subroutine STMMC (cont.)

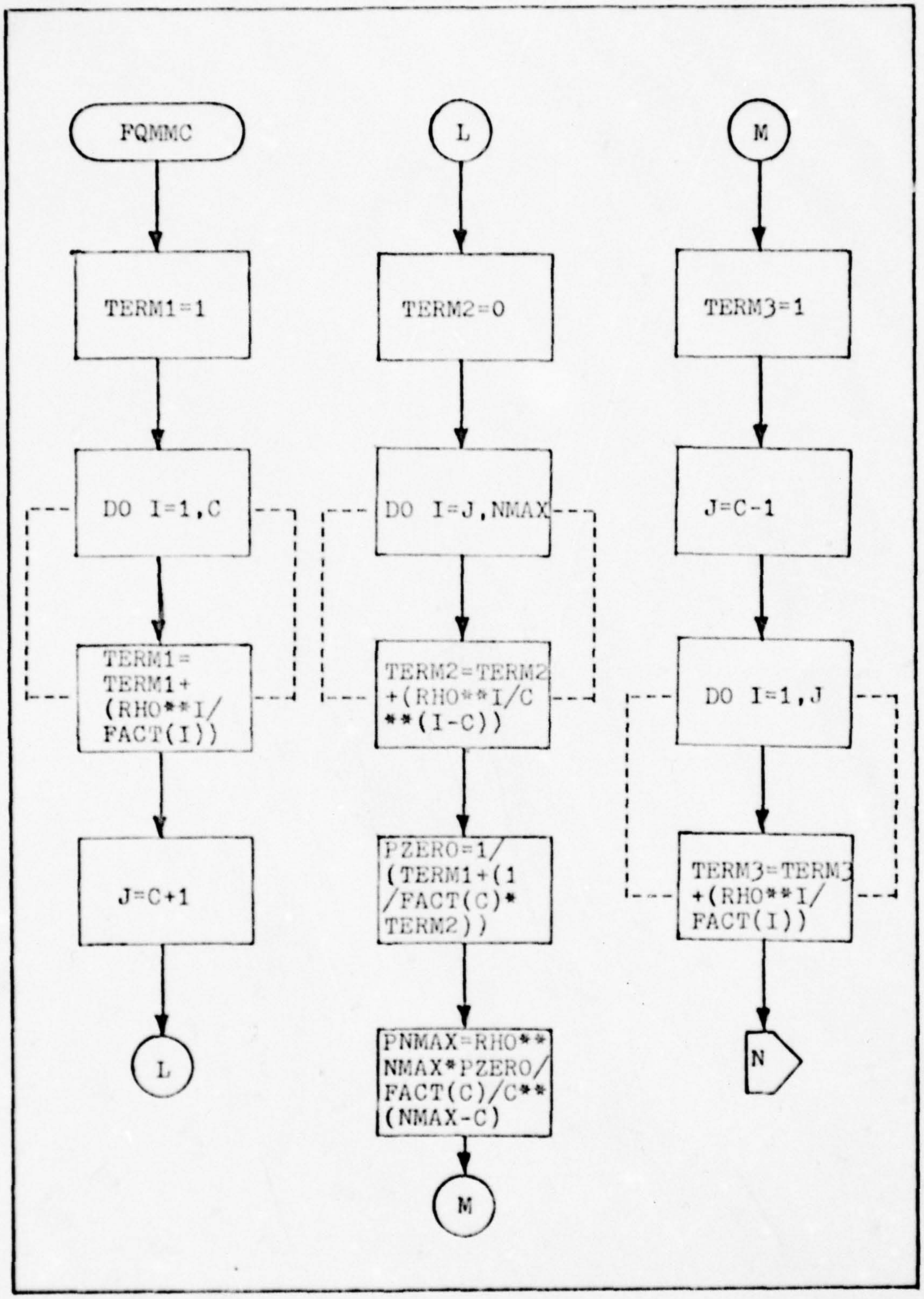


Figure 8. Subroutine FQMMC

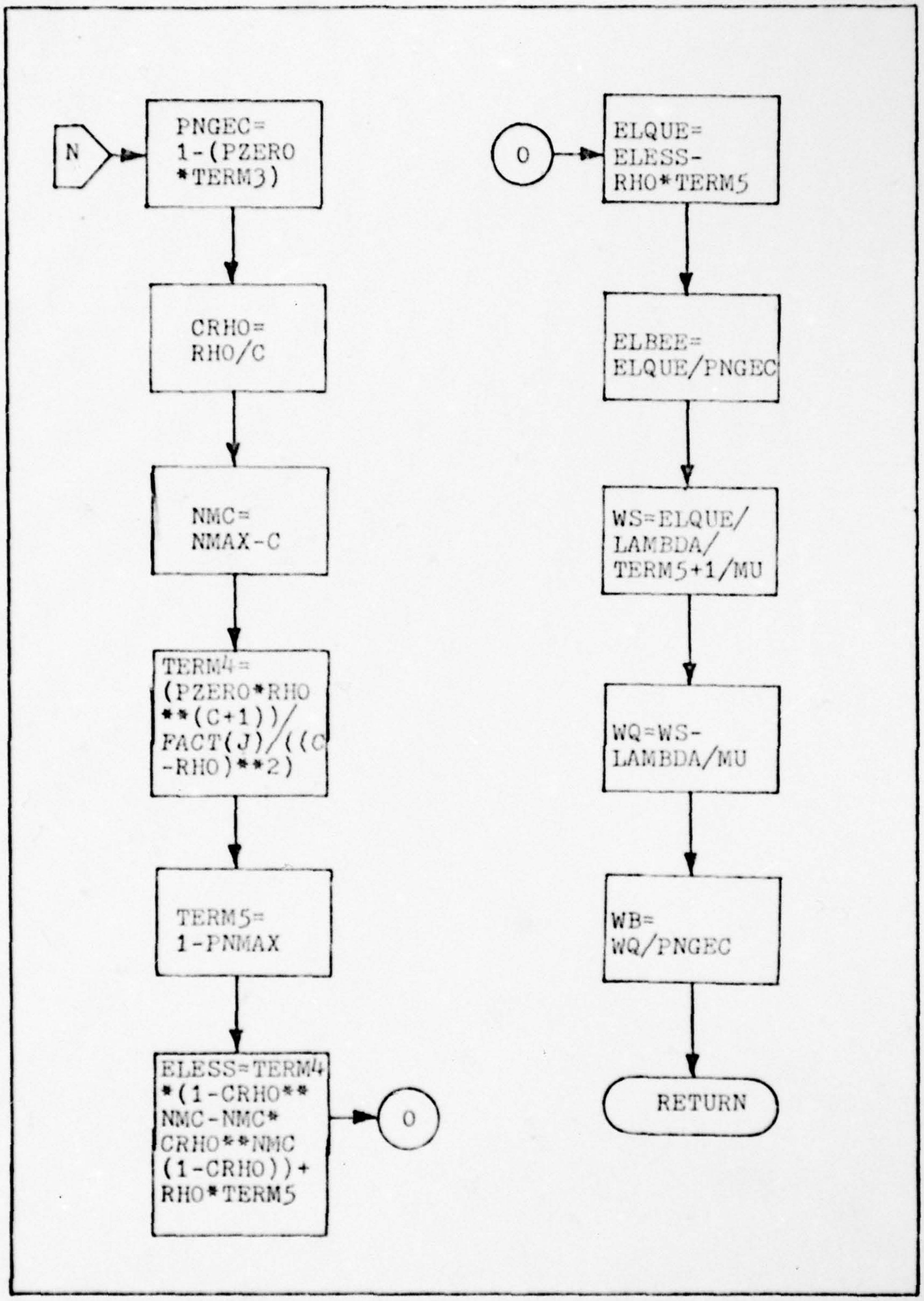


Figure 8. Subroutine FQMMC (cont.)

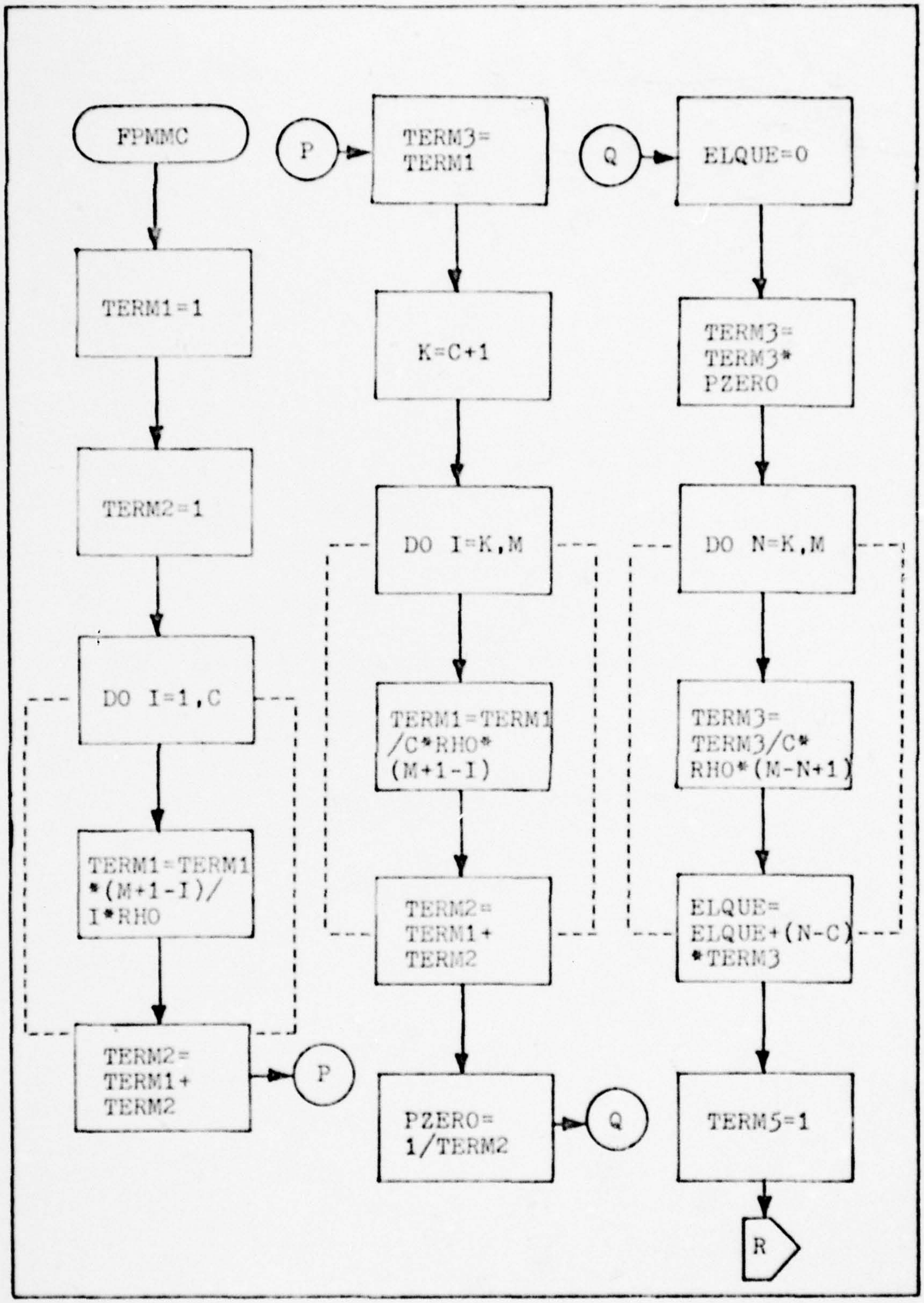


Figure 9. Subroutine FPMMC

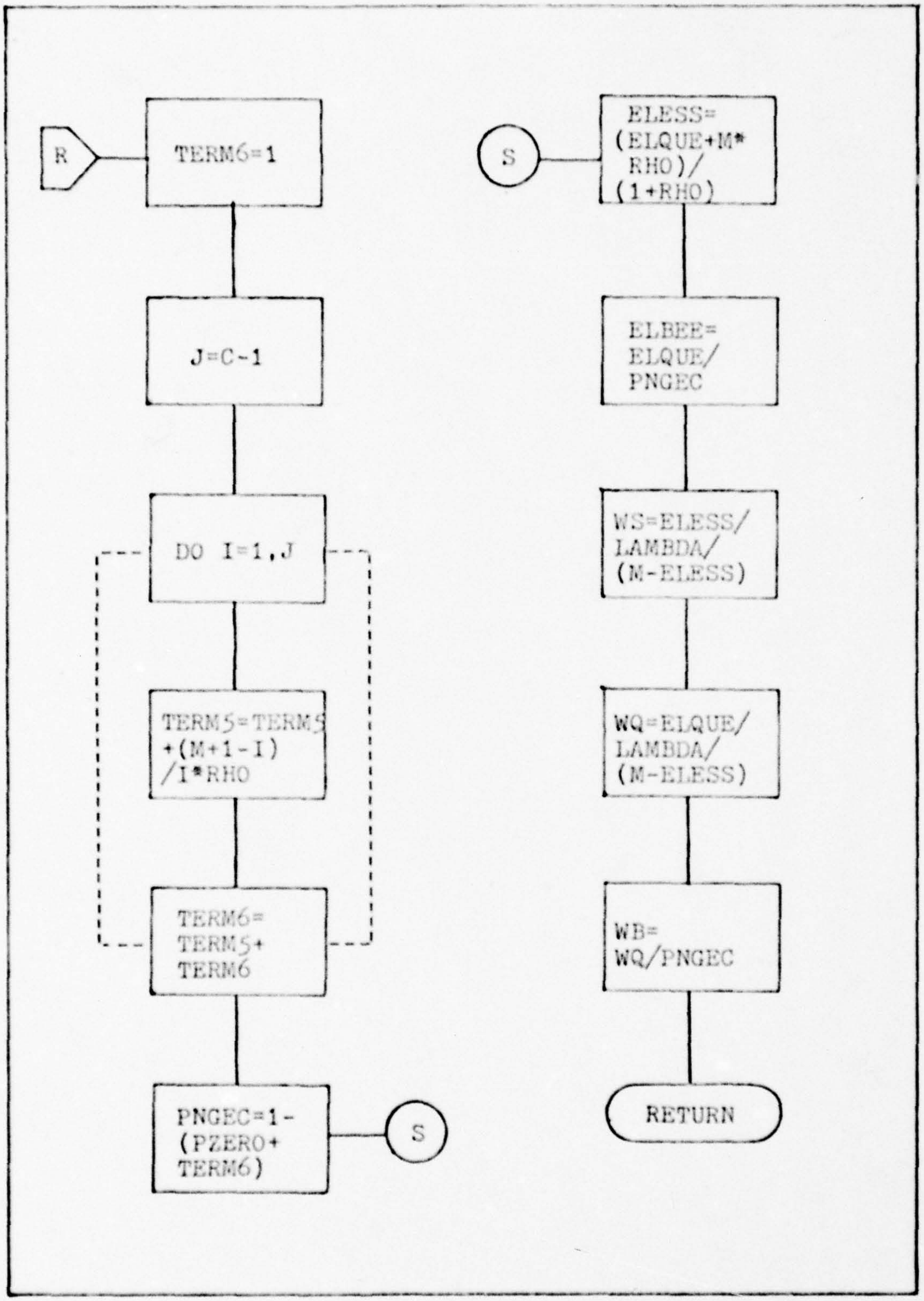


Figure 9. Subroutine FPMMC (cont.)

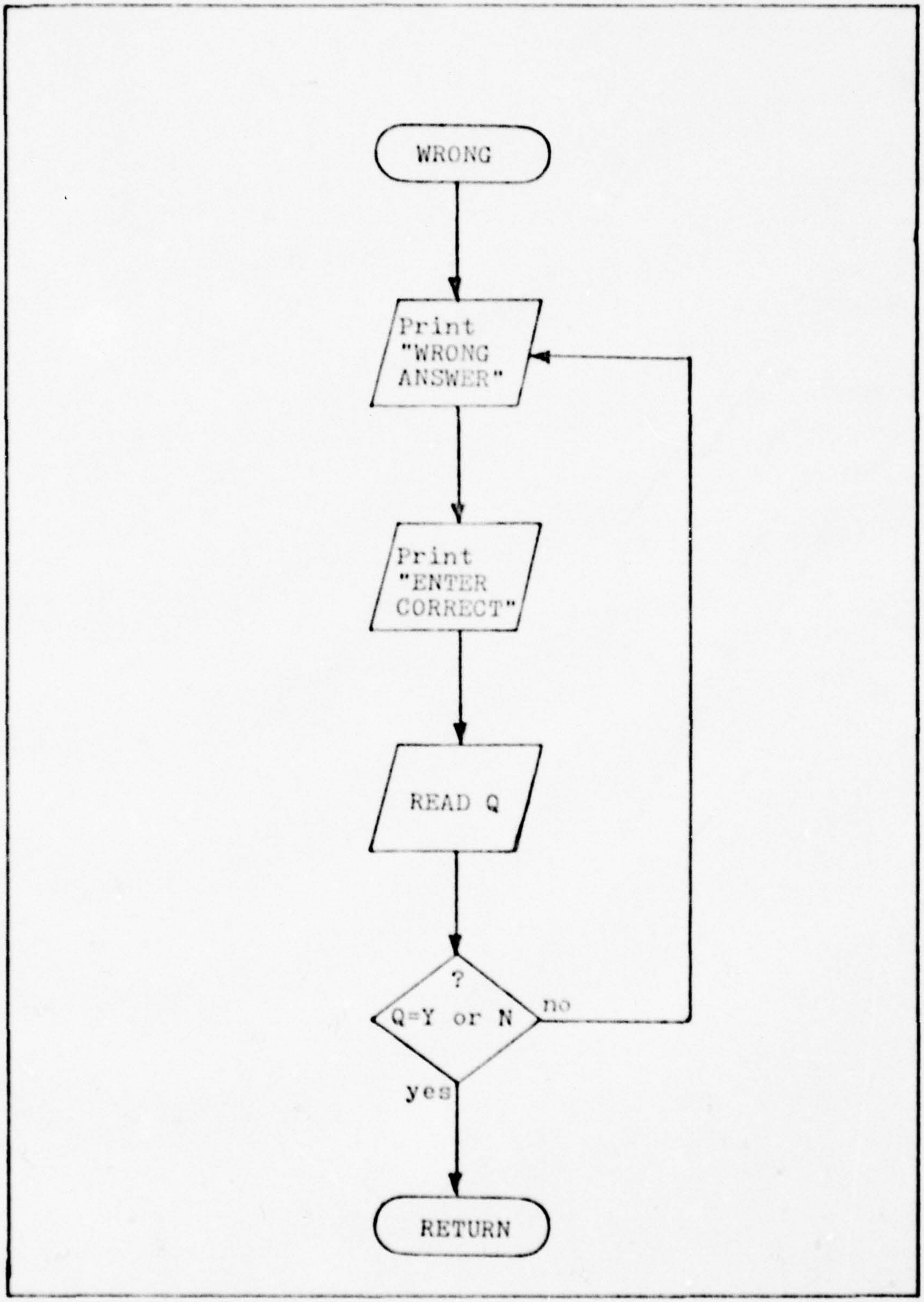


Figure 10. Subroutine WRONG

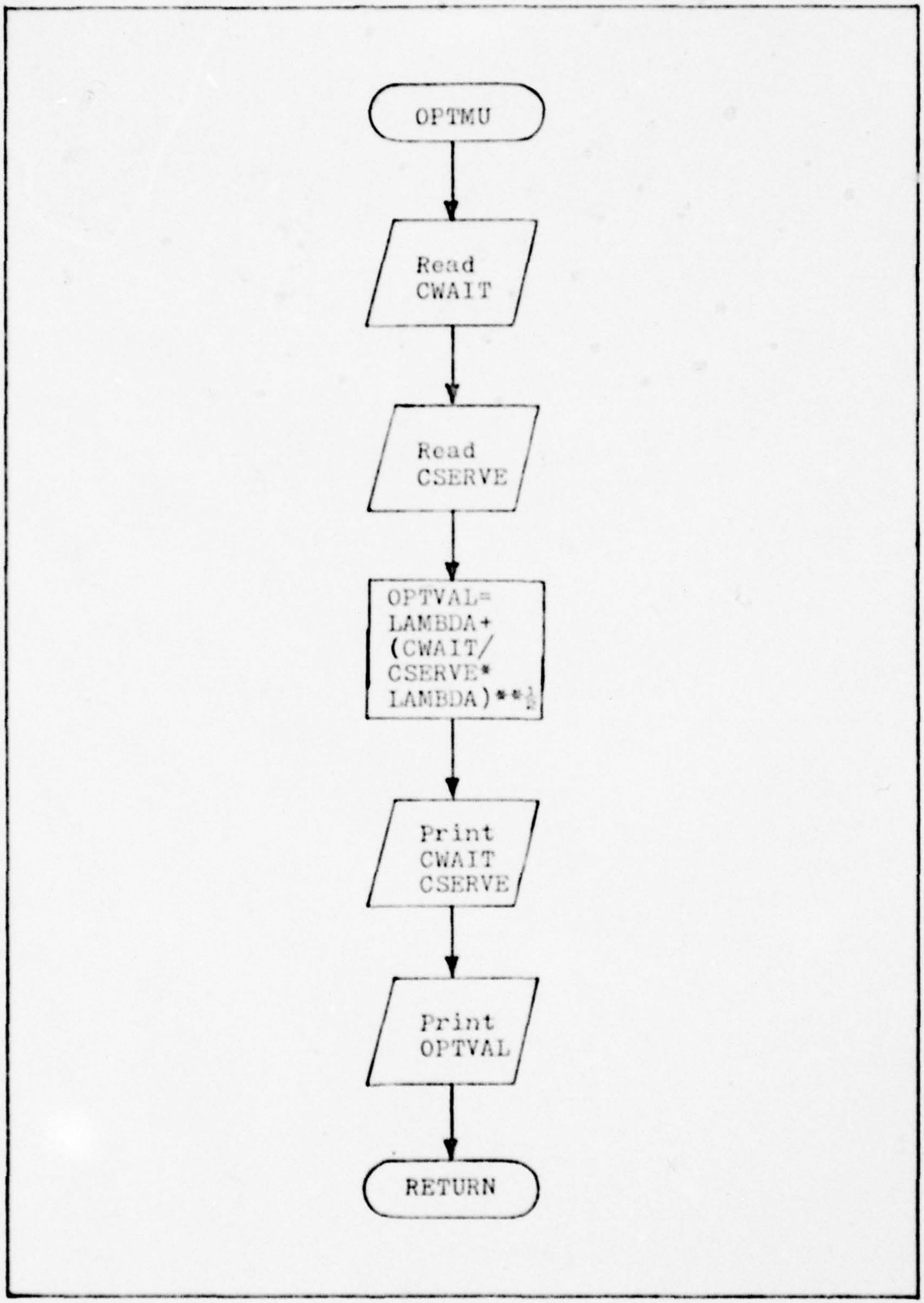


Figure 11. Subroutine OPTMU

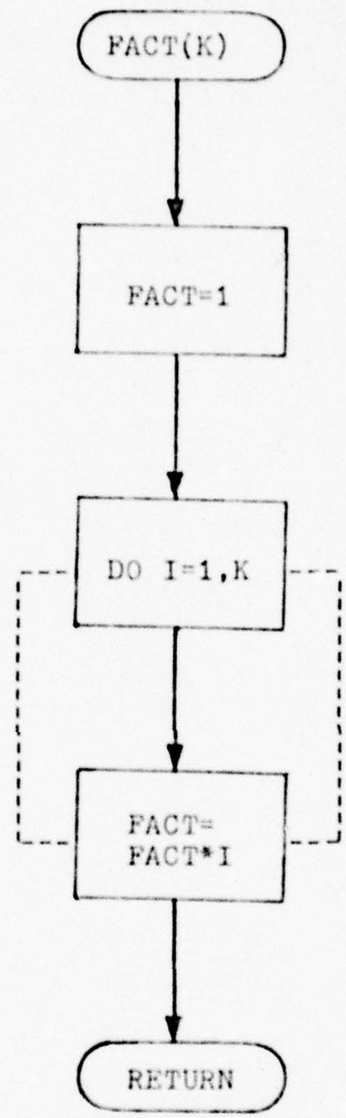


Figure 12. Subroutine FACT(K)

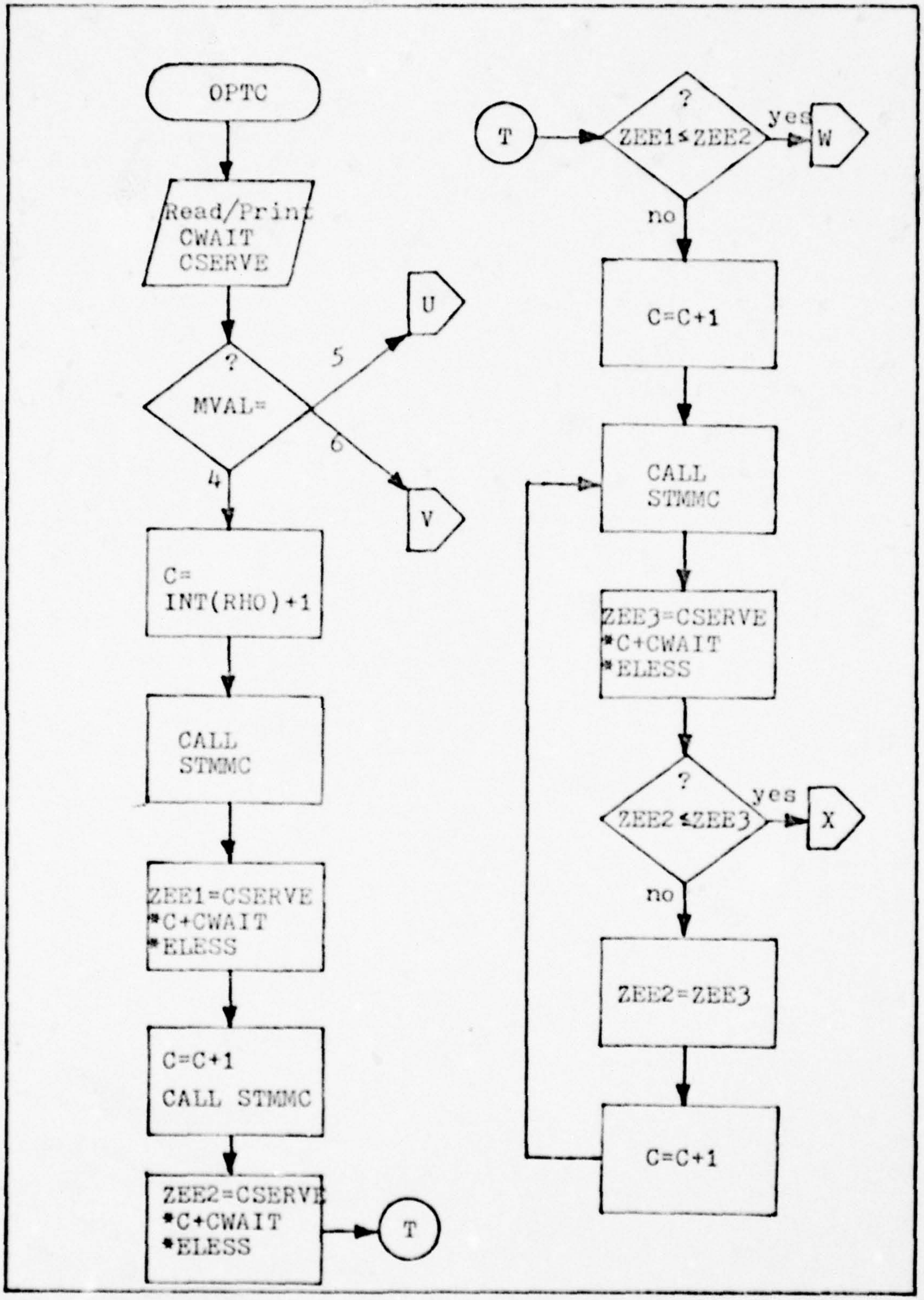


Figure 13. Subroutine OPTC

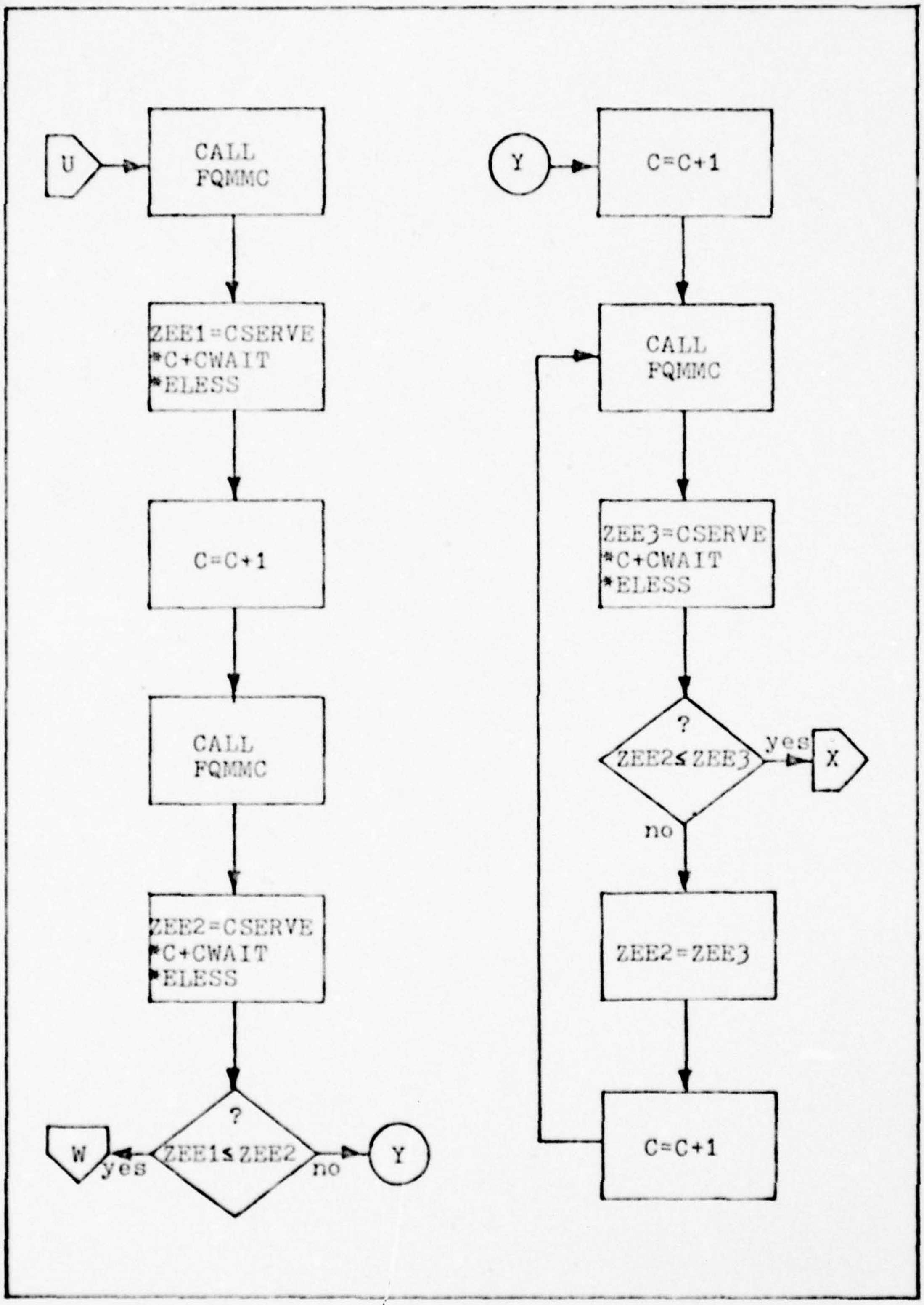


Figure 13. Subroutine OPTC (cont.)

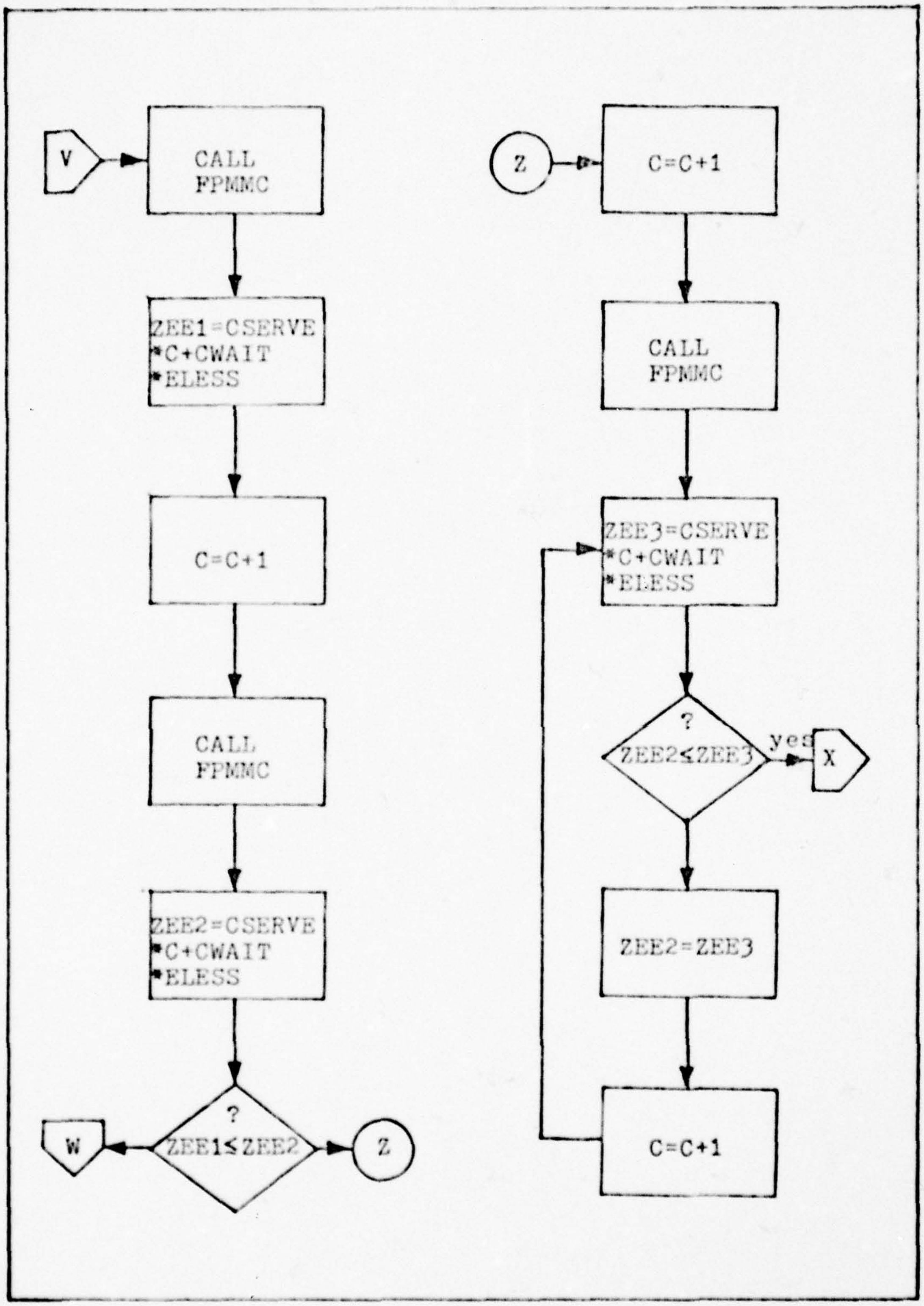


Figure 13. Subroutine OPTC (cont.)

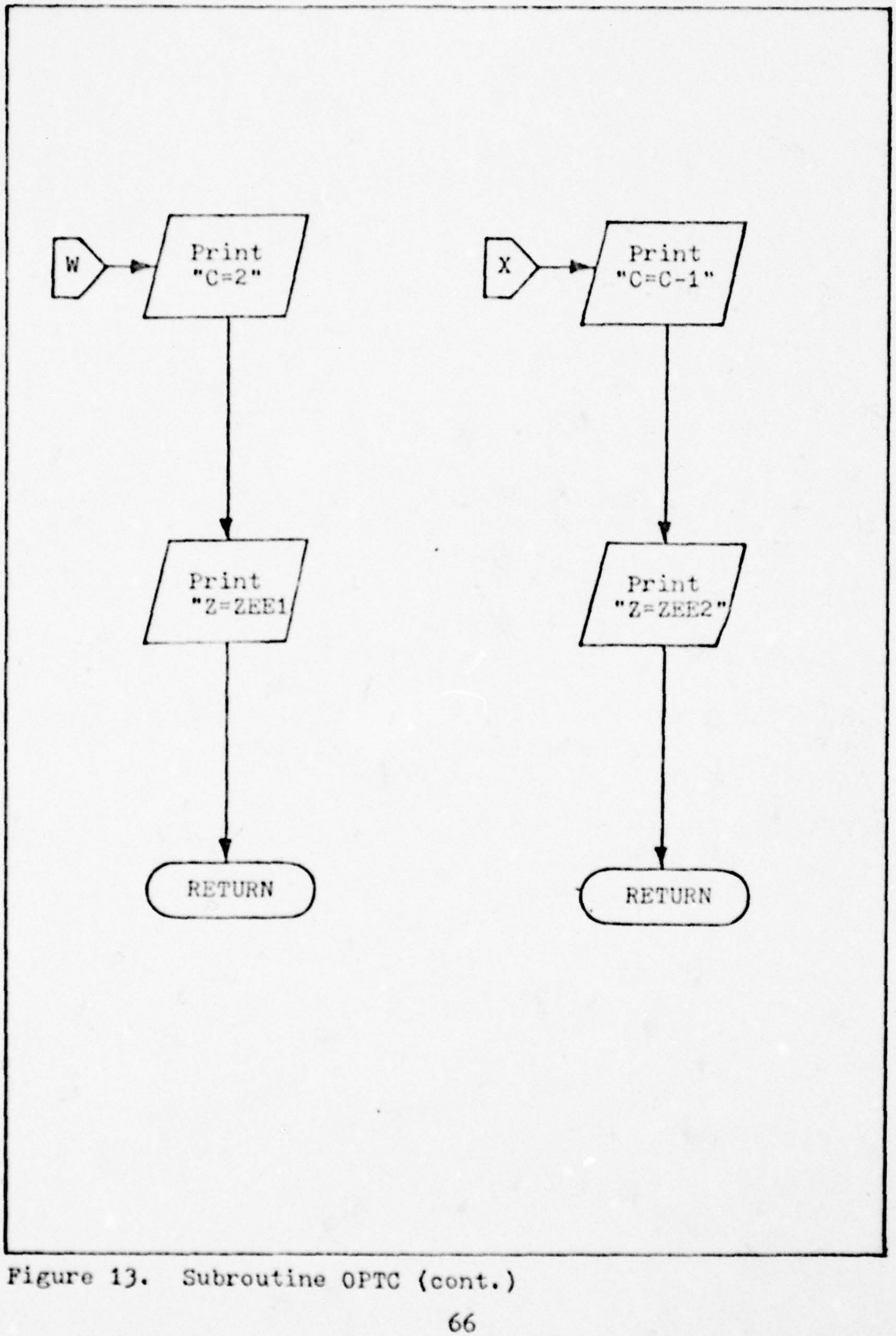


Figure 13. Subroutine OPTC (cont.)

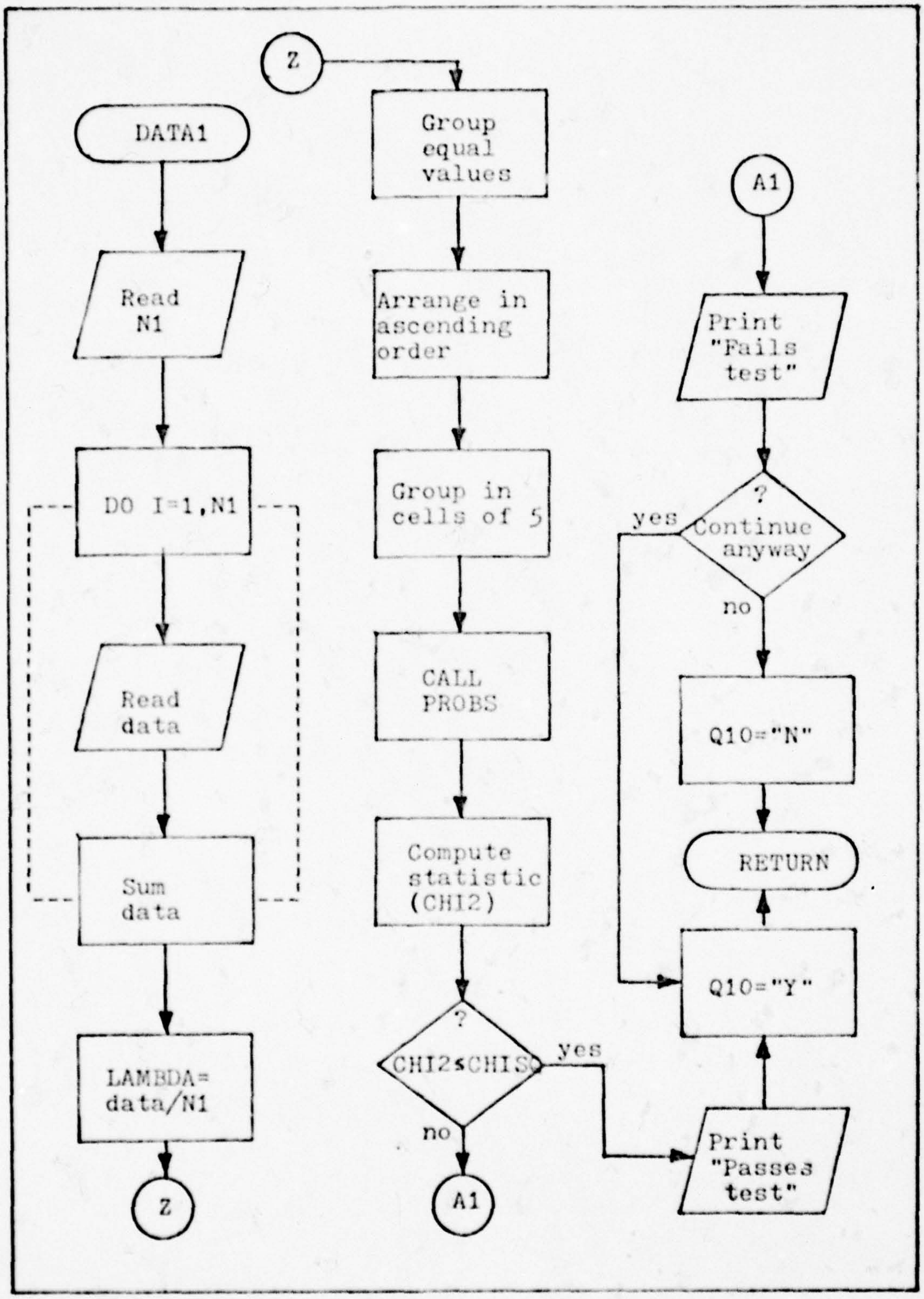


Figure 14. Subroutine DATA1

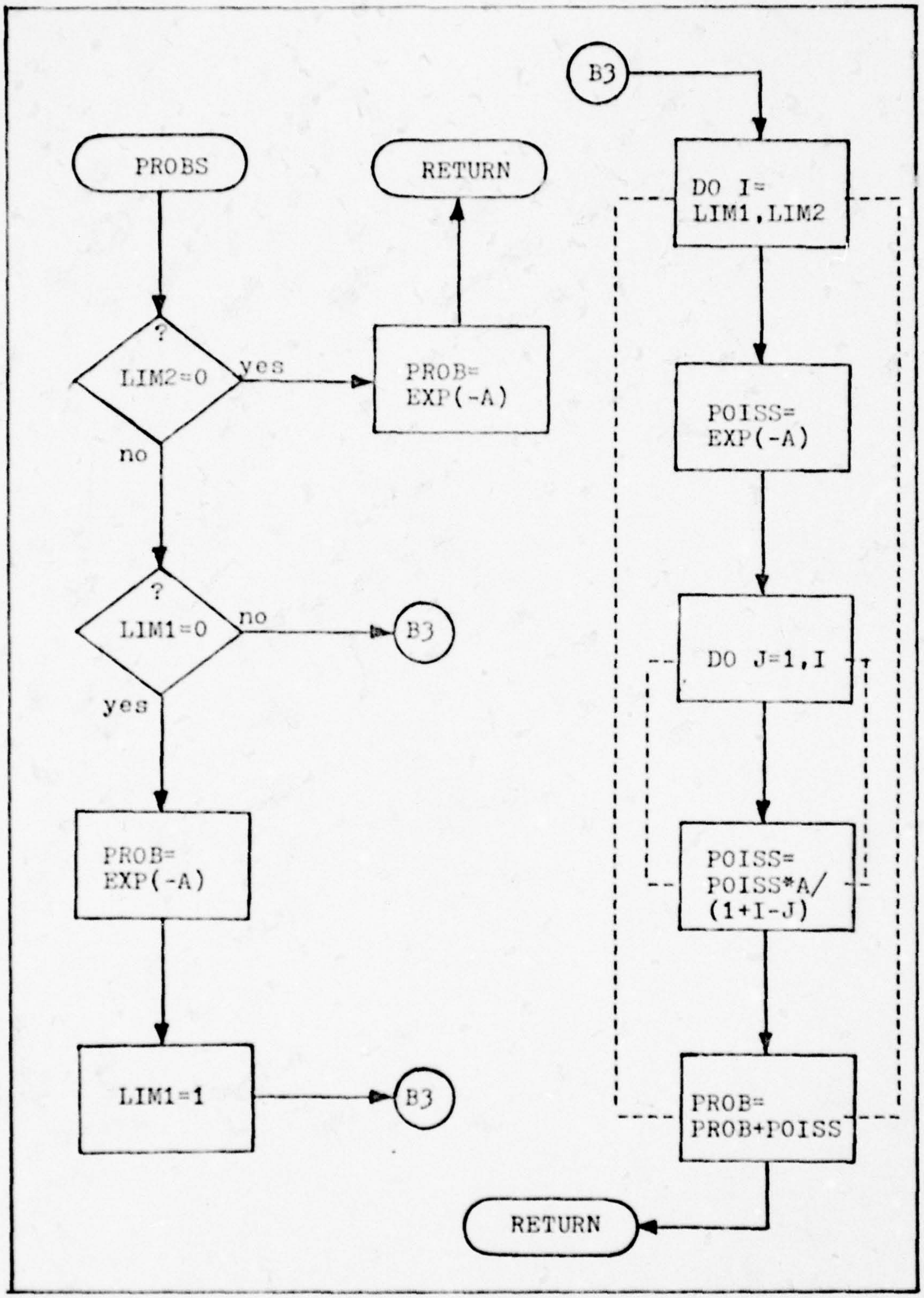


Figure 15. Subroutine PROBS

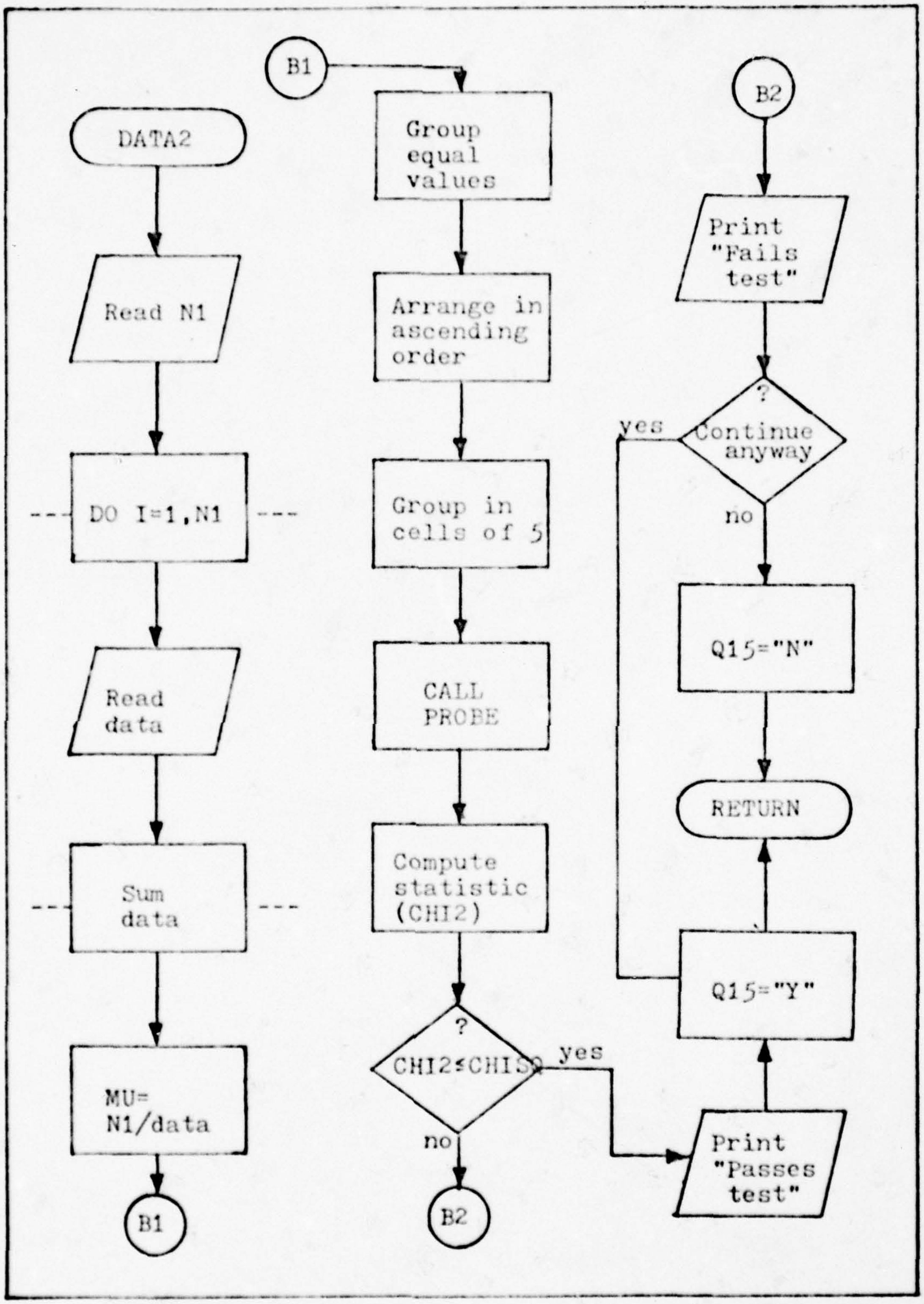


Figure 16. Subroutine DATA2

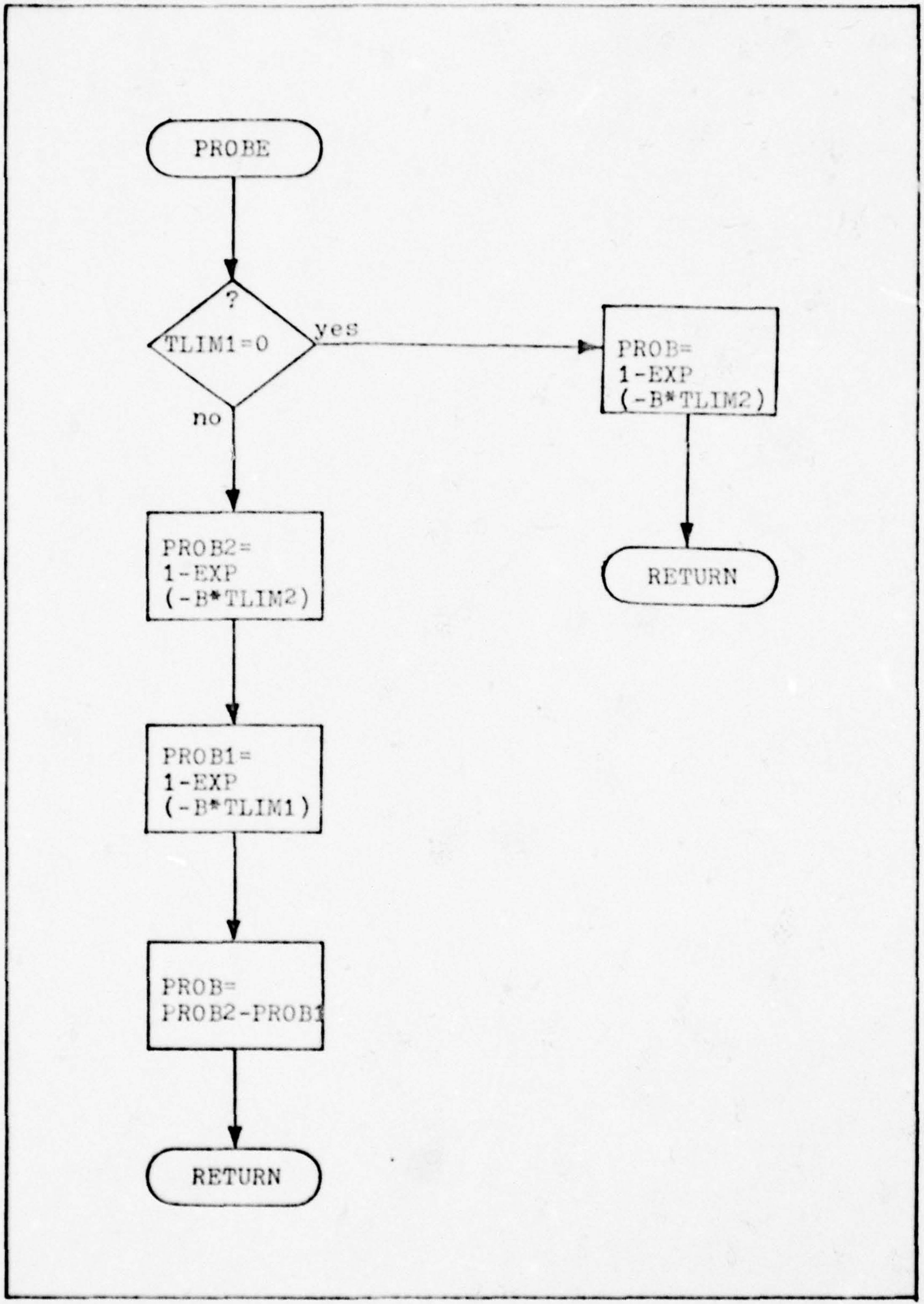


Figure 17. Subroutine PROBE

```
      PROGRAM MAIN(INPUT,OUTPUT)
      COMMON W,X,Y,Z,A,B,RHO,C,P7ERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      COMMON/D1/M,LIMQ,NMAX
      COMMON/D2/IND
      INTEGER D,Q1,Q3,Q4,Q5,Q6,Q7
  100 FORMAT(A1)
      W=0.
      X=.5
      Y=1.
      Z=2.
      PRINT*," "
      PRINT*,"DO YOU WISH TO REVIEW THE INSTRUCTIONS FOR"
      PRINT*,"THIS PROGRAM?  *"
      READ 100,Q17
      IF(Q17.NE."Y".AND.Q17.NE."N") CALL WRONG(Q17)
      IF(Q17.EQ."N") GO TO 90
      PRINT*," "
      PRINT*,"THIS COMPUTER PROGRAM IS DESIGNED TO COMPUTE STEADY-"
      PRINT*,"STATE CHARACTERISTICS FOR SERVICE SYSTEMS WHICH DEVELOP"
      PRINT*,"QUEUES (WAITING LINES) AND MEET THE FOLLOWING CONDITIONS:"
      PRINT*,"       1.  A STEADY STATE IS POSSIBLE."
      PRINT*,"       2.  ARRIVALS ARE INDEPENDENT AND BEHAVE"
      PRINT*,"           ACCORDING TO THE POISSON PROBABILITY"
      PRINT*,"           MASS FUNCTION."
      PRINT*,"       3.  SERVICE TIMES ARE DISTRIBUTED ACCORDING"
      PRINT*,"           TO THE EXPONENTIAL PROBABILITY DENSITY"
      PRINT*,"           FUNCTION."
      PRINT*,"       4.  WHEN THE POPULATION OF CUSTOMERS THAT"
      PRINT*,"           MAY BE USING THE SERVICE IS FINITE, THE"
      PRINT*,"           ELAPSED TIME BETWEEN LEAVING THE SYSTEM AND"
      PRINT*,"           RETURNING AGAIN FOR SERVICE FOR EACH CUSTOMER"
```

```
PRINT*,"           IS DISTRIBUTED ACCORDING TO THE EXPONENTIAL"
PRINT*,"           PROBABILITY DENSITY FUNCTION."
PRINT*," "
PRINT*,"THIS PROGRAM WILL CHECK THE FIRST CONDITION FOR"
PRINT*,"EVERY PROBLEM."
PRINT*," "
PRINT*,"IF SAMPLE DATA ON CUSTOMER ARRIVALS IS AVAILABLE, THIS"
PRINT*,"DATA MAY BE ENTERED AND IT WILL BE CHECKED FOR THE "
PRINT*,"SECOND CONDITION.  OTHERWISE, THE CUSTOMER ARRIVAL RATE"
PRINT*,"MAY BE ENTERED DIRECTLY AND IT WILL BE ASSUMED THAT THE"
PRINT*,"CONDITION IS SATISFIED."
PRINT*," "
PRINT*,"IF SAMPLE DATA ON SERVICE TIMES IS AVAILABLE, THIS"
PRINT*,"DATA MAY BE ENTERED AND IT WILL BE CHECKED FOR THE "
PRINT*,"THIRD CONDITION.  OTHERWISE THE SERVICE RATE MAY BE"
PRINT*,"ENTERED DIRECTLY AND IT WILL BE ASSUMED THAT THE"
PRINT*,"CONDITION IS SATISFIED."
PRINT*,"A CHECK FOR THE FOURTH CONDITION IS ALSO MADE"
PRINT*,"WHEN SAMPLE DATA FOR A FINITE POPULATION IS ENTERED."
PRINT*," "
PRINT*," "
PRINT*,"ALL RESPONSES YOU MUST MAKE WILL BE PRECEEDED BY SOME"
PRINT*,"TYPE OF PROMPTING STATEMENT.  RESPONSES ARE REQUIRED"
PRINT*,"WHENEVER THE PRINTER STOPS AFTER SUCH  A PROMPTING"
PRINT*,"STATEMENT; WHICH IS ALWAYS FOLLOWED BY AN ASTERISK (*)."
PRINT*,"YOU SHOULD TYPE YOUR RESPONSE IMMEDIATELY AFTER THE (*)"
PRINT*,"AND PUSH THE KEY MARKED CARRIAGE RETURN (RETURN,CR)."
PRINT*," "
PRINT*,"YOU WILL BE REQUIRED TO MAKE TWO TYPES OF RESPONSES:"
PRINT*," "
PRINT*,"THE FIRST TYPE IS THE ANSWER TO A 'YES' OR 'NO' QUESTION."
PRINT*,"IN THIS CASE AN ANSWER OTHER THAN 'YES' OR 'NO' WILL NOT"
PRINT*,"BE ACCEPTED.  (TRY IT!)"
PRINT*," "
PRINT*,"THE SECOND TYPE OF RESPONSE IS ENTRY OF SOME TYPE OF"
PRINT*,"NUMBER.  NO SPECIAL FORMAT IS REQUIRED.  IF THE NUMBER "
PRINT*,"NEEDS A DECIMAL POINT, PLACE IT WHERE IT BELONGS.  IF NO"
PRINT*,"POINT IS NEEDED, THEN DO NOT USE ONE.  "
PRINT*," "
PRINT*,"*NOTE* THIS PROGRAM IS NOT EQUIPPED TO CONVERT UNITS OF"
```

```
      PRINT*,"MEASURE.  THEREFORE, ALL DATA ENTERED SHOULD BE "
      PRINT*,"CONSISTENT IN UNITS OF MEASURE.  FOR EXAMPLE:  DO NOT "
      PRINT*,"ENTER DATA MEASURED IN HOURS ONE TIME AND DATA "
      PRINT*,"MEASURED IN SECONDS ANOTHER TIME.  USE ONE OR THE"
      PRINT*,"OTHER EXCLUSIVELY FOR ANY INDIVIDUAL PROBLEM."
      PRINT*," "
      PRINT*,"**IMPORTANT NOTE**  THIS PROGRAM MAY BE TERMINATED AT ANY"
      PRINT*,"  TIME BY ENTERING THE TWO CHARACTERS '%A'  THE PROGRAM"
      PRINT*,"MAY THEN BE STARTED AGAIN BY BEGINNING WITH "
      PRINT*,"THE REWIND COMMAND."
      PRINT*," "
90    PRINT*,"WHEN YOU ARE READY TO BEGIN, ENTER 'GO'.  *"
      READ 100,Q12
15    PRINT*,"THE MODELS AVAILABLE IN THIS PROGRAM ARE LISTED BELOW."
      PRINT*," "
      PRINT*,"*MODEL*                                   *CODE*"
      PRINT*,"STANDARD M/M/1 MODEL                        1"
      PRINT*,"FINITE QUEUE M/M/1 MODEL                    2"
      PRINT*,"FINITE POPULATION M/M/1 MODEL               3"
      PRINT*,"STANDARD M/M/C MODEL                        4"
      PRINT*,"FINITE QUEUE M/M/C MODEL                    5"
      PRINT*,"FINITE POPULATION M/M/C MODEL               6"
      PRINT*," "
      PRINT*," "
      PRINT*,"DO YOU KNOW WHICH MODEL YOU WANT TO USE?  *"
      READ 100,Q1
      PRINT*," "
      IF(Q1.NE."Y".AND.Q1.NE."N") CALL WRONG(Q1)
      IF(Q1.EQ."Y") GO TO 30
      PRINT*,"THE FOLLOWING SEQUENCE WILL HELP YOU CHOOSE THE APPROPRIAT
     1E MODEL."
95    PRINT*,"HOW MANY SERVERS ARE THERE IN THE SERVICE SYSTEM?  *"
      READ*,C
      IF(Q1.EQ."Y") GO TO 35
      IF(C.GT.1) MSIGN=3
      IF(C.EQ.1)MSIGN=0
      PRINT*,"IS THE LENGTH OF THE QUEUE LIMITED TO A FINITE NUMBER?  *"
      READ 100,Q3
      IF(Q3.NE."Y".AND.Q3.NE."N") CALL WRONG(Q3)
      IF(Q3.EQ."N") GO TO 10
```

```
      MVAL=2+MSIGN
96    PRINT*,"ENTER THE MAXIMUM LENGTH OF THE QUEUE.  *"
      READ*,LIMQ
      NMAX=LIMQ+C
      GO TO 40
10    PRINT*,"IS THE POPULATION THAT MIGHT USE THE SERVICE FINITE"
      PRINT*,"(USUALLY 30 UNITS OR LESS)?  *"
      READ 100,Q4
      IF(Q4.NE."Y".AND.Q4.NE."N") CALL WRONG(Q4)
      IF(Q4.EQ."N") GO TO 20
      MVAL=3+MSIGN
11    PRINT*,"ENTER THE SIZE OF THE FINITE POPULATION.  *"
      READ*,M
      GO TO 35
20    MVAL=1+MSIGN
      GO TO 40
30    PRINT*,"ENTER THE CODE FOR THE DESIRED MODEL.  *"
      READ*,MVAL
      IF(MVAL.LE.3) C=1
      IF(MVAL.GE.4) GO TO 95
35    IF(MVAL.EQ.2.OR.MVAL.EQ.5) GO TO 96
      IF(MVAL.EQ.3.OR.MVAL.EQ.6) GO TO 11
      GO TO 40
36    PRINT*,"FOR FINITE POPULATION PROBLEMS, CUSTOMER ARRIVAL DATA"
      PRINT*,"SHOULD BE IN TERMS OF INTERARRIVAL TIMES (THE TIME"
      PRINT*,"ELAPSED FROM WHEN A CUSTOMER LEAVES THE SERVICE"
      PRINT*,"SYSTEM TILL THAT SAME CUSTOMER ARRIVES AGAIN FOR SERVICE."
      PRINT*," "
      PRINT*,"DO YOU WISH TO ENTER SAMPLE DATA FOR INTERARRIVAL"
      PRINT*,"TIMES?  *"
      READ 100,Q25
      IF(Q25.NE."Y".AND.Q25.NE."N") CALL WRONG(Q25)
      IF(Q25.EQ."N") GO TO 37
      IND=6
      CALL DAT12(Q26)
      IF(Q26.EQ."N") GO TO 80
      GO TO 13
37    PRINT*,"ENTER THE MEAN (AVERAGE) INTERARRIVAL TIME"
      PRINT*,"PER INDIVIDUAL CUSTOMER (UNIT TIME PER CUSTOMER).  *"
      READ*,A1
```

```
      A=Y/A1
      GO TO 13
40    PRINT*," "
      PRINT*,"DO YOU WISH TO ENTER SAMPLE DATA"
      PRINT*,"FOR CUSTOMER ARRIVALS? *"
      READ 100,Q13
      IF(Q13.NE."Y".AND.Q13.NE."N") CALL WRONG(Q13)
      IF(Q13.EQ."N") GO TO 12
      CALL DATA1(Q10)
      IF(Q10.EQ."N") GO TO 80
      GO TO 13
12    PRINT*,"ENTER LAMBDA, THE MEAN (AVERAGE) ARRIVAL RATE"
      PRINT*,"(UNITS PER TIME PERIOD).  *"
      READ*,A
      PRINT*," "
13    PRINT*,"DO YOU WISH TO ENTER SAMPLE DATA FOR SERVICE TIMES?  *"
      READ 100,Q14
      IF(Q14.NE."Y".AND.Q14.NE."N") CALL WRONG(Q14)
      IF(Q14.EQ."N") GO TO 14
      IND=7
      CALL DATA2(Q15)
      IF(Q15.EQ."N") GO TO 80
      GO TO 38
14    PRINT*,"ENTER MU, THE MEAN (AVERAGE) SERVICE RATE (UNITS"
      PRINT*,"PER TIME PERIOD.  *"
      READ*,B
38    RHO=A/B
      PRINT*," "
      PRINT*," "
      PRINT*,"************************************************************"
      IF(MVAL.EQ.1) CALL STMM1
      IF(MVAL.EQ.2) CALL FQMM1(LIMQ,NMAX,PNMAX)
      IF(MVAL.EQ.3) CALL FPMM1(M)
      IF(MVAL.EQ.4) CALL STMMC(PNGEC)
      IF(MVAL.EQ.5) CALL FQMMC(LIMQ,NMAX,PNGEC,PNMAX)
      IF(MVAL.EQ.6) CALL FPMMC(M,PNGEC)
      PRINT*," "
      IF(MVAL.EQ.1) PRINT*,"YOU HAVE SELECTED THE STANDARD M/M/1 MODEL."
      IF(MVAL.EQ.2) PRINT*,"YOU HAVE SELECTED THE FINITE QUEUE M/M/1 MOD
     1EL."
```

```
      IF(MVAL.EQ.3) PRINT*,"YOU HAVE SELECTED THE FINITE POPULATION M/M/
     11 MODEL."
      IF(MVAL.EQ.4) PRINT*,"YOU HAVE SELECTED THE STANDARD M/M/C MODEL."
      IF(MVAL.EQ.5) PRINT*,"YOU HAVE SELECTED THE FINITE QUEUE M/M/C MOD
     1EL."
      IF(MVAL.EQ.6) PRINT*,"YOU HAVE SELECTED THE FINITE POPULATION M/M/
     1C MODEL."
      PRINT*," "
      PRINT*," "
      PRINT*,"THE FOLLOWING VALUES DESCRIBE THE STEADY-STATE"
      PRINT*,"OPERATION OF THE QUEUEING SYSTEM."
      PRINT*," "
      IF(MVAL.NE.3.AND.MVAL.NE.6) GO TO 41
      PRINT*,"MEAN ARRIVAL RATE PER CUSTOMER"
      PRINT*,"      (ARRIVALS PER TIME PERIOD):"
      PRINT*,"             LAMBDA = ",A
      PRINT*," "
      PRINT*,"EFFECTIVE MEAN ARRIVAL RATE FOR THE SYSTEM"
      PRINT*,"      (CUSTOMERS PER TIME PERIOD):"
      PRINT*,"             LAMBDA1 = ",A*(M-ELESS)
      PRINT*," "
      GO TO 42
41    PRINT*,"MEAN ARRIVAL RATE (UNITS PER TIME PERIOD):"
      PRINT*,"             LAMBDA = ",A
      PRINT*," "
42    PRINT*,"MEAN SERVICE RATE (UNITS PER TIME PERIOD):"
      PRINT*,"             MU = ",B
      PRINT*," "
      PRINT*,"TRAFFIC INTENSITY:"
      PRINT*,"             RHO = ",RHO
      PRINT*," "
      IF(MVAL.NE.2.AND.MVAL.NE.5) GO TO 50
      PRINT*,"MAXIMUM NUMBER ALLOWED IN THE SYSTEM:"
      PRINT*,"             N = ",NMAX
      PRINT*," "
      PRINT*,"PROBABILITY OF REJECTION DUE TO A FULL QUEUE:"
      PRINT*,"             P(N) = ",PNMAX
      PRINT*," "
50    IF(MVAL.NE.3.AND.MVAL.NE.6) GO TO 60
      PRINT*,"NUMBER OF UNITS IN FINITE POPULATION:"
```

```
      PRINT*,"            M = ",M
      PRINT*," "
60    PRINT*,"NUMBER OF SERVERS:"
      PRINT*,"            C = ",C
      IF(MVAL.NE.1.AND.MVAL.NE.4)GO TO 65
      IF(RHO.LT.C) GO TO 65
      PRINT*," "
      PRINT*,"THE ARRIVAL RATE EXCEEDS THE SERVICE RATE."
      PRINT*,"THIS SYSTEM WILL NOT REACH STEADY STATE."
      PRINT*," "
      PRINT*,"***************************************************"
      PRINT*," "
      GO TO 80
65    PRINT*," "
      PRINT*,"PROBABILITY OF THE SYSTEM BEING IDLE:"
      PRINT*,"            P(0) = ",PZERO
      PRINT*," "
      PRINT*,"STEADY-STATE MEAN NUMBER OF UNITS IN SYSTEM:"
      PRINT*,"            LS = ",ELESS
      PRINT*," "
      PRINT*,"STEADY-STATE MEAN NUMBER OF UNITS IN QUEUE:"
      PRINT*,"            LQ = ",ELQUE
      PRINT*," "
      PRINT*,"STEADY-STATE MEAN NUMBER OF UNITS IN QUEUE FOR BUSY SYSTEM
     1:"
      PRINT*,"            LB = ",ELBEE
      PRINT*," "
      PRINT*,"STEADY-STATE MEAN TIME IN SYSTEM:"
      PRINT*,"            WS = ",WS
      PRINT*," "
      PRINT*,"STEADY-STATE MEAN TIME IN QUEUE:"
      PRINT*,"            WQ = ",WQ
      PRINT*," "
      PRINT*,"STEADY-STATE MEAN TIME IN QUEUE FOR BUSY SYSTEM:"
      PRINT*,"            WB = ",WB
      PRINT*," "
      PRINT*,"**********************************************************"
      PRINT*," "
      IF(MVAL.EQ.4) GO TO 70
      [illegible] TO FIND [illegible] OPTIMAL SERVICE RATE (MU)"
```

```
      PRINT*,"BASED ON THE COST OF SERVICE AND THE COST OF WAITING?  *"
      READ 100,O5
      PRINT*," "
      IF(O5.NE."Y".AND.O5.NE."N") CALL WRONG(O5)
      IF(O5.EQ."Y") CALL OPTMU(A)
      GO TO 80
70    PRINT*,"DO YOU WANT TO FIND THE OPTIMAL NUMBER OF SERVERS (C)"
      PRINT*,"BASED ON THE COST OF SERVICE AND THE COST OF WAITING?  *"
      READ 100,O6
      PRINT*," "
      IF(O6.NE."Y".AND.O6.NE."N") CALL WRONG(O6)
      IF(O6.EQ."Y") CALL OPTC(MVAL)
80    PRINT*,"THIS PROBLEM IS COMPLETE.  DO YOU HAVE OTHER PROBLEMS"
      PRINT*,"FOR THIS PROGRAM TO SOLVE?  *"
      READ 100,O7
      IF(O7.NE."Y".AND.O7.NE."N") CALL WRONG(O7)
      IF(O7.EQ."Y") GO TO 90
      PRINT*," "
      PRINT*,"IT HAS BEEN NICE WORKING WITH YOU.  HAVE A NICE DAY."
      PRINT*," "
      STOP"END OF PROGRAM"
      END
      SUBROUTINE STMM1
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      IF(RHO.GE.1) RETURN
      PZERO=Y-RHO
      ELESS=A/(B-A)
      ELQUE=RHO*ELESS
      ELBEE=ELESS
      WS=ELESS/A
      WQ=WS*RHO
      WB=WS
      RETURN
      END
      SUBROUTINE FOMM1(LIMQ,NMAX,PNMAX)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      IF(A.EQ.B) GO TO 10
      PZERO=(Y-RHO)/(Y-RHO**(NMAX+Y))
      GO TO 20
10    PZERO=Y/(NMAX+Y)
```

```
20    IF(A.EQ.B) GO TO 30
      ELESS=(RHO/(Y-RHO))-(((NMAX+Y)*RHO**(NMAX+Y))/(Y-RHO**(NMAX+Y)))
      GO TO 40
30    ELESS=NMAX/Z
40    ELQUE=ELESS-(Y-PZERO)
      ELBEE=ELQUE/(Y-PZERO)
      PNMAX=PZERO*RHO**NMAX
      WS=ELQUE/(A*(Y-PNMAX))+Y/B
      WQ=WS-(Y/B)
      WB=WQ/(Y-PZERO)
      RETURN
      END
      SUBROUTINE FRMM1(M)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      J=M-Y
      TERM1=M*RHO
      DPZERO=Y+TERM1
      DO 10 I=1,J
      TERM1=TERM1*(M-I)*RHO
10    DPZERO=DPZERO+TERM1
      PZERO=Y/DPZERO
      PBUSY=Y-PZERO
      ELESS=M-(PBUSY/RHO)
      ELQUE=M-((A+B)*PBUSY/A)
      ELBEE=ELQUE/PBUSY
      WS=(M/B/PBUSY)-(Y/A)
      WQ=((M/PBUSY)-((A+B)/A))/B
      WB=WQ/PBUSY
      RETURN
      END
      SUBROUTINE STMMC(PNGEC)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      INTEGER C
      IF(RHO.GE.C) RETURN
      J=C-Y
      TERM1=Y
      TERM2=Y
      TERM3=RHO**C/(Y-RHO/C)
      DO 10 I=1,J
      TERM1=TERM1*RHO/I
```

```
      TERM2=TERM1+TERM2
   10 TERM3=TERM3/(I+Y)
      PZERO=Y/(TERM2+TERM3)
      PNGEC=RHO**C*B*C*PZERO/(B*C-A)
      DO 20 I=2,C
   20 PNGEC=PNGEC/I
      ELQUE=RHO**(C+Y)*PZERO/(C-RHO)**Z
      DO 30 I=1,J
   30 ELQUE=ELQUE/I
      ELESS=ELQUE+RHO
      ELBEE=ELQUE/PNGEC
      WS=(ELQUE/A)+(Y/B)
      WQ=ELQUE/A
      WB=WQ/PNGEC
      RETURN
      END
      SUBROUTINE FQMMC(LIMQ,NMAX,PNGEC,PNMAX)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      INTEGER C
      TERM1=Y
      DO 10 I=1,C
   10 TERM1=TERM1+(RHO**I/FACT(I))
      J=C+Y
      TERM2=W
      DO 20 I=J,NMAX
      IF(TERM2.GE.1.8447E+17) GO TO 40
   20 TERM2=TERM2+(RHO**I/C**(I-C))
      PZERO=Y/(TERM1+(Y/FACT(C)*TERM2))
      PNMAX=RHO**NMAX*PZERO/FACT(C)/C**(NMAX-C)
      TERM3=Y
      J=C-Y
      DO 30 I=1,J
   30 TERM3=TERM3+(RHO**I/FACT(I))
      PNGEC=Y-(PZERO*TERM3)
      CRHO=RHO/C
      NMC=NMAX-C
      TERM4=(PZERO*RHO**(C+Y))/FACT(J)/((C-RHO)**Z)
      TERM5=1-PNMAX
      ELESS=TERM4*(Y-CRHO**NMC-NMC*CRHO**NMC*(Y-CRHO))+RHO*TERM5
      ELQUE=ELESS-RHO*TERM5
```

```
      ELBEE=ELQUE/PNGEC
      WS=ELQUE/A/TERM5+(Y/B)
      WQ=WS-(Y/B)
      WB=WQ/PNGEC
      RETURN
40    PZERO=W
      PNMAX=W
      PNGEC=Y
      ELESS=RHO
      ELQUE=W
      ELBEE=W
      WS=Y/B
      WQ=W
      WB=W
      RETURN
      END
      SUBROUTINE FPMMC(M,PNGEC)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      INTEGER C,W
      TERM1=Y
      TERM2=Y
      DO 10 I=1,C
      TERM1=TERM1*(M+Y-I)/I*RHO
10    TERM2=TERM1+TERM2
      TERM3=TERM1
      K=C+Y
      DO 20 I=K,M
      TERM1=TERM1/C*RHO*(M+Y-I)
20    TERM2=TERM2+TERM1
      PZERO=Y/TERM2
      ELQUE=W
      TERM3=TERM3*PZERO
      DO 30 N=K,M
      TERM3=TERM3/C*RHO*(M-N+Y)
30    ELQUE=ELQUE+(N-C)*TERM3
      TERM5=Y
      J=C-Y
      DO 40 I=1,J
40    TERM5=TERM5+(M+Y-I)/I*RHO
      PNGEC=Y-(PZERO*TERM5)
```

```
      ELESS=(ELQUE+M*RHO)/(Y+RHO)
      ELBEE=ELQUE/PNGEC
      WS=ELESS/A/(M-ELESS)
      WQ=ELQUE/A/(M-ELESS)
      WB=WQ/PNGEC
      RETURN
      END
      SUBROUTINE WRONG(Q)
      INTEGER Q
100   FORMAT(A1)
10    PRINT*,"THE RESPONSE TO THE PREVIOUS QUESTION MUST BE 'YES'"
      PRINT*,"OR 'NO'.  YOU HAVE ENTERED AN INCORRECT RESPONSE."
      PRINT*,"PLEASE ENTER THE CORRECT RESPONSE AT THIS TIME.   *"
      READ 100,Q
      IF(Q.EQ."Y".OR.Q.EQ."N")RETURN
      GO TO 10
      END
      SUBROUTINE OPTMU(A)
      X=.5
      PRINT*,"ENTER THE COST OF WAITING PER CUSTOMER PER UNIT TIME.   *$"
      READ*,CWAIT
      PRINT*," "
      PRINT*,"ENTER THE MARGINAL SCOST OF SERVICE PER CUSTOMER.   *$"
      READ*,CSERVE
      OPTVAL=A+(CWAIT/CSERVE*A)**X
      PRINT*," "
      PRINT*,"**********************************************************"
      PRINT*," "
      PRINT*,"THE COST OF WAITING PER CUSTOMER PER UNIT TIME IS:"
      PRINT*,"            CW = $",CWAIT
      PRINT*,"THE MARAGINAL COST OF SERVICE PER CUSTOMER IS:"
      PRINT*,"            CS = $",CSERVE
      PRINT*," "
      PRINT*,"THE OPTIMAL MEAN SERVICE RATE (UNITS PER TIME PERIOD) IS:"
      PRINT*,"            MU = ",OPTVAL
      RETURN
      END
      FUNCTION FACT(K)
      FACT=1
      DO 10 I=1,K
```

```
10    FACT=FACT*I
      RETURN
      END
      SUBROUTINE OPTC(MVAL)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      COMMON/C1/M,LIMO,NMAX
      INTEGER C,W
      PRINT*,"ENTER THE COST OF WAITING PER CUSTOMER PER UNIT TIME.  *$"
      READ*,CWAIT
      PRINT*," "
      PRINT*,"ENTER THE COST PER SERVER PER UNIT TIME.  *$"
      READ*,CSERVE
      PRINT*," "
      PRINT*,"****************************************************************"
      PRINT*," "
      PRINT*,"THE COST OF WAITING PER CUSTOMER PER UNIT TIME IS:"
      PRINT*,"              CW = $",CWAIT
      PRINT*," "
      PRINT*,"THE COST PER SERVER PER UNIT TIME IS:"
      PRINT*,"              CS = $",CSERVE
      PRINT*," "
      PRINT*," "
      C=2
      IF(MVAL.EQ.5) GO TO 20
      IF(MVAL.EQ.6) GO TO 40
      C=INT(RHO)+Y
      CALL STMMC(PNGEC)
      ZEE1=CSERVE*C+CWAIT*ELESS
      C=C+Y
      CALL STMMC(PNGEC)
      ZEE2=CSERVE*C+CWAIT*ELESS
      IF(ZEE1.LE.ZEE2) GO TO 60
      C=C+Y
10    CALL STMMC(PNGEC)
      ZEE3=CSERVE*C+CWAIT*ELESS
      IF(ZEE2.LE.ZEE3) GO TO 70
      ZEE2=ZEE3
      C=C+Y
      GO TO 10
20    CALL FOMMC(LIMO,NMAX,PNGEC,PNMAX)
```

```
      ZEE1=CSERVE*C+CWAIT*ELESS
      C=C+Y
      CALL FOMMC(LIMQ,NMAX,PNGEC,PNMAX)
      ZEE2=CSERVE*C+CWAIT*ELESS
      IF(ZEE1.LE.ZEE2) GO TO 60
      C=C+Y
30    CALL FOMMC(LIMQ,NMAX,PNGEC,PNMAX)
      ZEE3=CSERVE*C+CWAIT*ELESS
      IF(ZEE2.LE.ZEE3) GO TO 70
      ZEE2=ZEE3
      C=C+Y
      GO TO 30
40    CALL FPMMC(M,PNGEC)
      ZEE1=CSERVE*C+CWAIT*ELESS
      C=C+Y
      CALL FPMMC(M,PNGEC)
      ZEE2=CSERVE*C+CWAIT*ELESS
      IF(ZEE1.LE.ZEE2) GO TO 60
      C=C+Y
50    CALL FPMMC(M,PNGEC)
      ZEE3=CSERVE*C+CWAIT*ELESS
      IF(ZEE2.LE.ZEE3) GO TO 70
      ZEE2=ZEE3
      C=C+Y
      GO TO 50
60    PRINT*,"THE OPTIMAL NUMBER OF SERVERS FOR THIS SYSTEM IS:"
      PRINT*,"              C = 2"
      PRINT*," "
      PRINT*," "
      PRINT*," "
      PRINT*,"THE COST OF THE SYSTEM IN DOLLARS PER TIME PERIOD IS:"
      PRINT*,"              Z = $",ZEE1
      PRINT*," "
      PRINT*,"*******************************************************"
      RETURN
70    PRINT*,"THE OPTIMAL NUMBER OF SERVERS FOR THIS SYSTEM IS:"
      PRINT*,"              C = ",C-1
      PRINT*," "
      PRINT*,"THE COST OF THE SYSTEM IN DOLLARS PER TIME PERIOD IS:"
      PRINT*,"              Z = $",ZEE2
```

```
      PRINT*,"*********************************************************************"
      RETURN
      END
      SUBROUTINE DATA1(O10)
C THIS SUBROUTINE COLLECTS DATA AND CHECKS FOR
C POISSON DISTRIBUTION USING THE CHI SQUARE
C GOODNESS-OF-FIT TEST.
      DIMENSION FREQ(100),ARRIV(100),CHISQ(37)
      COMMON W,X,Y,Z,A,B,RHO,C,FZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
100   FORMAT(A1)
      CHISQ(1)=6.63490
      CHISQ(2)=9.20134
      CHISQ(3)=11.3499
      CHISQ(4)=13.2767
      CHISQ(5)=15.0863
      CHISQ(6)=16.8119
      CHISQ(7)=18.4753
      CHISQ(8)=20.0902
      CHISQ(9)=21.6660
      CHISQ(10)=23.2093
      CHISQ(11)=24.7250
      CHISQ(12)=26.2170
      CHISQ(13)=27.6883
      CHISQ(14)=29.1413
      CHISQ(15)=30.5779
      CHISQ(16)=31.9999
      CHISQ(17)=33.4087
      CHISQ(18)=34.8053
      CHISQ(19)=36.1908
      CHISQ(20)=37.5662
      CHISQ(21)=38.9321
      CHISQ(22)=40.2894
      CHISQ(23)=41.6384
      CHISQ(24)=42.9798
      CHISQ(25)=44.3141
      CHISQ(26)=45.6417
      CHISQ(27)=46.9630
      CHISQ(28)=48.2782
      CHISQ(29)=49.5879
      CHISQ(3)=50.8922
```

AD-A076 982 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 5/10
INTERACTIVE PROGRAMMING FOR MANAGEMENT AIDS TO DECISION-MAKING:--ETC(U)
SEP 79 P HAMILTON
UNCLASSIFIED AFIT/GSM/SM/79S-6 NL

2 OF 2
AD
A076982

END
DATE
FILMED
12 79
DDC

```
      CHISQ(31)=63.6907
      CHISQ(32)=76.1539
      CHISQ(33)=88.3794
      CHISQ(34)=100.425
      CHISQ(35)=112.329
      CHISQ(36)=124.116
      CHISQ(37)=135.807
      PRINT*," "
      PRINT*,"YOU HAVE CHOSEN TO ENTER SAMPLE DATA."
      PRINT*,"LAMBDA, THE MEAN ARRIVAL RATE WILL BE COMPUTED FOR YOU."
      PRINT*,"DATA WILL BE CHECKED TO DETERMINE IF IT MEETS THE"
      PRINT*,"ASSUMPTIONS NECESSARY FOR USE IN THESE MODELS."
      PRINT*," "
      PRINT*,"ENTER THE NUMBER OF EQUAL TIME PERIODS OBSERVED."
      PRINT*,"(15 MINIMUM, 100 MAXIMUM)."
      READ*,N1
      PRINT*," "
      PRINT*,N1," TIME PERIODS WILL BE LISTED."
      PRINT*,"AFTER EACH PERIOD, ENTER THE NUMBER OF ARRIVALS"
      PRINT*,"OBSERVED DURING THAT PERIOD."
      PRINT*," "
      PRINT*," "
      ASUM=W
      DO 10 I=1,N1
            PRINT*,"ARIVALS DURING PERIOD ",I,"  *"
            READ*,ARRIV(I)
            FREQ(I)=W
10          ASUM=ASUM+ARRIV(I)
      A=ASUM/N1
      IF(ASUM.LT.15.) GO TO 91
ARRIV ARRAY WILL CONTAIN THE NUMBER OF TIMES
THAT A PARTICULAR FREQUENCY WAS OBSERVED.
      I=W
      DO 40 J=1,N1
            IF(ARRIV(J).EQ.(-1)) GO TO 40
            I=I+Y
            FREQ(I)=FREQ(I)+Y
            ARRIV(I)=ARRIV(J)
            LEFT=J+Y
            DO 30 K=LEFT,N1
```

```
               IF(ARRIV(J).NE.ARRIV(K)) GO TO 30
               ARRIV(K)=-Y
               FREQ(I)=FREQ(I)+Y
30           CONTINUE
          NCELL=I
40        CONTINUE
ARRANGE FREQ AND ARRIV IN ASCENDING ORDER OF ARRIV
     L=NCELL-Y
     DO 50 I=1,L
          DO 45 J=I,NCELL
               IF(ARRIV(I).LE.ARRIV(J)) GO TO 45
               SAVE1=ARRIV(I)
               SAVE2=FREQ(I)
               ARRIV(I)=ARRIV(J)
               FREQ(I)=FREQ(J)
               ARRIV(J)=SAVE1
               FREQ(J)=SAVE2
45             CONTINUE
50        CONTINUE
COMPRESS CELLS TO SIZE OF 5 OR GREATER
FIGURE EXPECTED VALUES, AND CHI STATISTIC.
INITIALIZE VALUES
     CHI2=W
     PSUM=W
     CSUM=W
     CSIZE=W
     LIM2=-1
     NDF=W
     DO 70 I=1,NCELL
          CSIZE=CSIZE+FREQ(I)
          IF(CSIZE.LT.5.) GO TO 70
          NDF=NDF+Y
          IF((N1-CSUM-CSIZE).LT.5.) GO TO 60
          LIM1=LIM2+Y
          LIM2=ARRIV(I)
          CALL PROBS(LIM1,LIM2,PROB)
          PSUM=PSUM+PROB
          EVAL=N1*PROB
          CHI2=CHI2+(CSIZE-EVAL)**2/EVAL
          CSUM=CSUM+CSIZE
```

```
      [illegible]) GO TO 80
      NDF=NDF-30
      NDF=NDF/10
      NDF=NDF+31
80    IF(CHI2.LE.CHISQ(NDF)) GO TO 90
81    PRINT*,"**************************************************************"
      PRINT*,"BASED ON THE SAMPLE DATA PRESENTED, YOUR ARRIVAL"
      PRINT*,"DISTRIBUTION *FAILS* TO MEET THE ASSMPTIONS OF "
      PRINT*,"THESE MODELS.  (ARRIVALS ARE NOT DISTRIBUTED "
      PRINT*,"EXPONENTIALLY)."
92    PRINT*,"THE DESCRIPTIVE CHARACTERISTICS PRESENTED BY THESE "
      PRINT*,"MODELS MAY NOT BE VALID FOR YOUR SERVICE SYSTEM."
      PRINT*," "
      PRINT*,"DO YOU WANT TO CONTINUE WITH THE PROBLEM ANYWAY?  *"
      READ 100,Q10
      IF(Q10.NE."Y".AND.Q10.NE."N") CALL WRONG(Q10)
      PRINT*,"**************************************************************"
      RETURN
91    PRINT*,"**************************************************************"
      PRINT*,"YOU HAVE NOT PRESENTED A SUFFICIENT NUMBER OF"
      PRINT*,"OBSERVATIONS TO DETERMINE WHETHER OR NOT YOUR ARRIVAL"
      PRINT*,"DISTRIBUTION MEETS THE ASSUMPTIONS OF THESE MODELS."
      GO TO 92
90    PRINT*,"**************************************************************"
      PRINT*,"BASED ON THE SAMPLE DATA PRESENTED, YOUR CUSTOMER"
      PRINT*,"ARRIVAL DISTRIBUTION MEETS THE ASSUMPTIONS OF"
      PRINT*,"THESE MODELS."
      PRINT*,"**************************************************************"
      Q10="Y"
      RETURN
```

```
      SUBROUTINE POISSN(LIM1,LIM2,PROB)
      COMMON W,X,Y,Z,A,B,RHO,C,P7ERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
TO COMPUTE THE PROBABILITY OF AN ARRIVAL WITHIN A CELL
DEFINED BY LIM1 AND LIM2,BASED ON POISSON FUNCTION.
      PROB=W
      IF(LIM2.EQ.0) GO TO 30
      IF(LIM1.EQ.0) GO TO 50
40    DO 20 I=K,LIM2
           POISS=EXP(-A)
           DO 10 J=1,I
10              POISS=POISS*A/(I+Y-J)
20         PROB=PROB+POISS
      RETURN
50    PROB=EXP(-A)
      K=1
      GO TO 40
30    PROB=EXP(-A)
      RETURN
      END
      SUBROUTINE DATA2(Q15)
THIS SUBROUTINE COLLECTS DATA AND CHECKS FOR EXPONENTIAL
DISTRIBUTION USING THE CHI SQUARE GOODNESS-OF-FIT TEST.
      DIMENSION FREQ(100),STIM(100),CHISQ(37)
      COMMON W,X,Y,Z,A,B,RHO,C,P7ERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      COMMON/C2/IND
100   FORMAT(A1)
      CHISQ(1)=6.63490
      CHISQ(2)=9.21034
      CHISQ(3)=11.3449
      CHISQ(4)=13.2767
      CHISQ(5)=15.0863
      CHISQ(6)=16.8119
      CHISQ(7)=18.4753
      CHISQ(8)=20.0902
      CHISQ(9)=21.6660
      CHISQ(10)=23.2093
      CHISQ(11)=24.7250
      CHISQ(12)=26.2170
      CHISQ(13)=27.6883
```

```
CHISQ(14)=29.1413
CHISQ(15)=30.5779
CHISQ(16)=31.9999
CHISQ(17)=33.4087
CHISQ(18)=34.8053
CHISQ(19)=36.1908
CHISQ(20)=37.5662
CHISQ(21)=38.9321
CHISQ(22)=40.2894
CHISQ(23)=41.6384
CHISQ(24)=42.9798
CHISQ(25)=44.3141
CHISQ(26)=45.6417
CHISQ(27)=46.9630
CHISQ(28)=48.2782
CHISQ(29)=49.5879
CHISQ(30)=50.8922
CHISQ(31)=63.6907
CHISQ(32)=76.1539
CHISQ(33)=88.3794
CHISQ(34)=100.425
CHISQ(35)=112.329
CHISQ(36)=124.116
CHISQ(37)=135.807
PRINT*," "
PRINT*,"YOU HAVE CHOSEN TO ENTER SAMPLE DATA."
IF(IND.EQ.6) GO TO 11
PRINT*,"MU, THE MEAN SERVICE RATE WILL BE COMPUTED FOR YOU."
PRINT*,"DATA WILL BE CHECKED TO DETERMINE IF IT MEETS THE"
PRINT*,"ASSUMPTIONS NECESSARY FOR USE IN THESE MODELS."
PRINT*," "
PRINT*,"ENTER THE NUMBER OF CUSTOMERS SERVICED WHILE DATA"
PRINT*,"WAS GATHERED (MAXIMUM 100)."
READ*,N1
PRINT*," "
PRINT*,N1," CUSTOMER NUMBERS WILL BE PRINTED."
PRINT*,"AFTER EACH CUSTOMER NUMBER, ENTER THE AMOUNT OF TIME"
PRINT*,"NEEDED TO SERVICE EACH CUSTOMER."
PRINT*," "
PRINT*," "
```

```
      TSUM=W
      DO 10 I=1,N1
           PRINT*,"CUSTOMER #",I,"  *"
           READ*,STIM(I)
           FREQ(I)=W
10         TSUM=TSUM+STIM(I)
      B=N1/TSUM
      D=N1/TSUM
      GO TO 13
11    PRINT*,"LAMBDA, THE MEAN CUSTOMER ARRIVAL RATE, WILL BE "
      PRINT*,"COMPUTED FOR YOU.  DATA WILL BE CHECKED TO"
      PRINT*,"DETERMINE IF IT MEETS THE ASSUMPTIONS NECESSARY"
      PRINT*,"FOR USE IN THESE MODELS."
      PRINT*," "
      PRINT*,"ENTER THE NUMBER OF INTERARRIVAL TIMES OBSERVED.  *"
      READ*,N1
      PRINT*," "
      PRINT*,N1," INTERARRIVAL OBSERVATION NUMBERS WILL BE"
      PRINT*,"PRINTED.  AFTER EACH OBSERVATION NUMBER, ENTER"
      PRINT*,"THE INTERARRIVAL TIME OBSERVED."
      PRINT*," "
      TSUM=W
      DO 12 I=1,N1
           PRINT*,"OBSERVATION #",I,"  *"
           READ*,STIM(I)
           FREQ(I)=W
12         TSUM=TSUM+STIM(I)
      A=N1/TSUM
      D=N1/TSUM
13    IF(N1.LT.15) GO TO 92
STIM ARRAY WILL CONTAIN THE TIME TO SERVICE A CUSTOMER.
FREQ ARRAY WILL CONTAIN THE FREQUENCY THAT THE SERVICE TIME
OCCURRED
      L=W
      DO 40 I=1,N1
           IF(STIM(I).EQ.-1.) GO TO 40
           L=L+Y
           STIM(L)=STIM(I)
           FREQ(L)=Y
           J=I+Y
```

```
          IF(I.EQ.N1) GO TO 50
          DO 30 K=J,N1
               IF(STIM(K).EQ.-1.) GO TO 30
               IF(STIM(I).NE.STIM(K)) GO TO 30
               FREQ(L)=FREQ(L)+Y
               STIM(K)=-1.
30        CONTINUE
          NCELL=L
40        CONTINUE
ARRANGE FREQ AND STIM IN ASCENDING ORDER OF STIM VALUES
50   L=NCELL-Y
     DO 70 I=1,L
          DO 60 J=I,NCELL
               IF(STIM(I).LE.STIM(J)) GO TO 60
               SAVE1=STIM(I)
               SAVE2=FREQ(I)
               STIM(I)=STIM(J)
               FREQ(I)=FREQ(J)
               STIM(J)=SAVE1
               FREQ(J)=SAVE2
60             CONTINUE
70        CONTINUE
COMPRESS DATA INTO CELLS OF 5 OR GREATER.
COMPUTE EXPECTED VALUES AND CHI STATISTIC.
     CHI2=W
     PSUM=W
     CSUM=W
     CSIZE=W
     TLIM2=0.
     NDF=W
     DO 80 I=1,NCELL
          CSIZE=CSIZE+FREQ(I)
          IF(CSIZE.LT.5.) GO TO 80
          NDF=NDF+Y
          IF((N1-CSUM-CSIZE).LT.5.) GO TO 71
          TLIM1=TLIM2
          TLIM2=STIM(I)
          CALL PROBE(TLIM1,TLIM2,PROB,D)
          PSUM=PSUM+PROB
          EVAL=N1*PROB
```

```
            CHI2=CHI2+(CSIZE-EVAL)**2/EVAL
            CSUM=CSUM+CSIZE
            CSIZE=W
            GO TO 80
71          IF(NDF.LT.3) GO TO 91
            PROB=Y-PSUM
            CSIZE=N1-CSUM
            EVAL=N1*PROB
            CHI2=CHI2+(CSIZE-EVAL)**2/EVAL
            GO TO 81
80          CONTINUE
81    NDF=NDF-7
      IF(NDF.LE.30) GO TO 90
      NDF=NDF-30
      NDF=NDF/10
      NDF=NDF+31
90    IF(CHI2.GT.CHISQ(NDF)) GO TO 91
      PRINT*,"************************************************************"
      PRINT*,"BASED ON THE SAMPLE DATA PRESENTED, YOUR"
      PRINT*,"TIME DISTRIBUTION MEETS THE ASSUMPTIONS OF THESE MODELS"
      Q15="Y"
      PRINT*,"************************************************************"
      RETURN
91    PRINT*,"********************************************************"
      PRINT*,"BASED ON THE SAMPLE DATA PRESENTED, YOUR"
      PRINT*,"DISTRIBUTION *FAILS* TO MEET THE ASSUMPTIONS OF THESE"
      PRINT*,"MODELS.  (SERVICE TIMES ARE NOT EXPONENTIALLY"
      PRINT*,"DISTRIBUTED.)"
      GO TO 93
92    PRINT*,"********************************************************"
      PRINT*,"YOU HAVE NOT PRESENTED A SUFFICIENT NUMBER OF "
      PRINT*,"OBSERVATIONS TO DETERMINE WHETHER OR NOT YOUR"
      PRINT*,"TIME DISTRIBUTION MEETS THE ASSUMPTIONS OF THESE MODELS"
93    PRINT*,"THE DESCRIPTIVE CHARACTERISTICS PRESENTED BY THESE"
      PRINT*,"MAY NOT BE VALID FOR YOUR SERVICE SYSTEM."
      PRINT*,"DO YOU WANT TO CONTINUE WITH THE MODEL ANYWAY?  *"
      READ 100,Q15
      IF(Q15.NE."Y".AND.Q15.NE."N") CALL WRONG(Q15)
      PRINT*,"************************************************************"
      RETURN
```

```
      END
      SUBROUTINE PROBE(TLIM1,TLIM2,PROB,D)
      COMMON W,X,Y,Z,A,B,RHO,C,PZERO,ELESS,ELQUE,ELBEE,WS,WQ,WB
      IF(TLIM1.EQ.0.) GO TO 10
      PROB2=Y-EXP(-D*TLIM2)
      PROB1=Y-EXP(-D*TLIM1)
      PROB=PROB2-PROB1
      RETURN
10    PROB=Y-EXP(-D*TLIM2)
      RETURN
      END

***********        PZHOOED   //// END  OF LIST ////
```

APPENDIX C

INSTRUCTIONS FOR QUEUE1

QUEUE1: AN INTERACTIVE PROGRAM

Program Description

QUEUE1 is a computer program designed to aid managers with decision-making efforts concerning waiting lines (queues) and associated service systems. QUEUE1 incorporates mathematical models which describe a certain class of service systems characterized by the following assumptions:

1. Customer arrivals to the service system may be reasonably described by the Poisson probability mass function.
2. Customers are "patient" and "well-behaved"; that is, they will enter a waiting line regardless of its length and will not leave a waiting line once they have entered it.
3. Customers waiting in line are served on a "first-come, first-served" basis.
4. All servers are identical, and service times may be approximated by the exponential probability density function.

If it is uncertain that customer arrival and service time distributions meet these assumptions, QUEUE1 will perform a test on observations to check for compliance. QUEUE1 accepts inputs that define the service system in question and generates output that describes the steady-state operation of the service system. Refer to the examples

included in these instructions for a list of these characteristics. For single server systems, QUEUE1 can compute an optimum service rate which will result in the lowest cost of operating that system. For multiple server systems, QUEUE1 will compute the number of servers which results in the lowest operating cost. The models and equations used in QUEUE1 and discussed in these instructions may be referenced in Principals of Operations Research for Management (Budnick, Mojena, and Vollman; Richard D. Irwin, Inc., 1977).

Definitions

Understanding the following definitions will help in executing QUEUE1.

A QUEUE is a waiting line. It is made up of customers waiting for some type of service to be performed by a server. A customer is not necessarily a person. A customer is any unit requiring the service in question. A customer may be a letter on a conveyor belt waiting to be processed. The size of the queue does not include those customers already being serviced.

The ARRIVAL RATE is a measure of the frequency with which customers arrive at the service system. Arrival rates are expressed in number of customers per time period. In general, the arrival rate refers to how many customers arrive during a time period. However, in finite population problems, the arrival rate refers to how often an individual member of the population arrives during a time period.

The SERVICE RATE is a measure of the speed which a server can process customers. In QUEUE1 the service rate must be expressed in the same units as the arrival rate, customer units that can be serviced per time period.

A FINITE QUEUE SYSTEM is one in which the length of the queue is limited to a certain length. When the queue is full, additional customers who arrive are not allowed to enter the queue. They are said to be rejected. The queue for a drive-in banking facility might be limited if the

off street waiting space only had room for a small number of cars. Population size does not constitute a queue limit.

A FINITE POPULATION SYSTEM is one in which the number of customers that may need service is limited to a certain number. For example, office workers using the same copying machine represent a finite population. Populations of 30 or more can usually be considered infinite.

Preparation

Before initiating QUEUE1, gather the following information:

1. Number of servers in the system.

2. Maximum number of customers allowed in the queue at any one time (if applicable).

3. The size of the customer population that might be using the queue (normally, a population size of 30 or more is considered infinite).

4. The marginal cost of service per customer, if the optimal service rate is to be computed (optional and applies to single server systems only).

5. The cost per server per unit time, if the optimal number of servers is to be computed.

6. The cost of having a customer wait per unit time, for either of the two optimization cases.

7. Raw observation data. Example 1 shows the kind of raw observation data that may be entered and the method of entry. Mean service rates and mean arrival rates (averages) may be entered instead of observation data. This is shown in example 2.

Care should be taken to insure that all inputs agree in unit dimensions. That is, if a service rate is expressed in units per hour, then arrival rates must be expressed in units per hour (not units per second or minute).

Remote terminal Operation

QUEUE1 requires the active participation of the user. The following set of instructions will enable the user to access QUEUE1 on a remote terminal in the AFIT computer facilities. These instructions are not intended to address all operating procedures for the AFIT remote terminals. They are adequate only to direct the use of QUEUE1. A more detailed explanation can be found in the current edition of the ASD Computer Center INTERCOM Guide (Ref 4). The following instructions are adapted from that publication.

Terminal Hookup

Different types of terminals are used at AFIT. It may be necessary to modify the following steps slightly, depending upon the control/keyboard configuration of the specific terminal being used.

1. Check to see that the terminal is plugged in.

2. If the DATA PHONE is the removable type, check to see if it is plugged in.

3. Check the controls on the front panel and under the top cover. The settings should be

Power	ON
On Line	ON
Speed	HIGH
Duplex	HALF
Line Feed	1

4. Push the TALK/CLEAR button on the DATA PHONE.

5. Dial the appropriate 300 BAUD number (currently 5180 or 5159).

6. When you hear the high pitched tone through the receiver, depress the DATA button on the DATA PHONE and replace the receiver.

7. Press the carriage return key on the terminal keyboard (marked CARRIAGE or CR).

8. You will receive a message on the printer. The last line will be "PLEASE LOGIN". If you do not receive this message, then the computer is down or there is a problem with the terminal.

LOGIN Procedure

The LOGIN command has the following format

LOGIN,xnnnnnn,xxxxxxxxxx,TID

The paramters are positional and required. The first parameter is the user's problem number. The second parameter is the user's assigned INTERCOM password. It may be up to ten positions in length and may contain any combination of letters or digits. The problem number and password may be assigned by your course instructor or the AFIT computer center. The third parameter is the three position terminal identifier. The TID appears on a label on the terminal. A typical LOGIN command is shown in the examples. Note that commas must separate each parameter. Once typed, the command is transmitted by pressing the carriage return key. If you make an error on the first attempt, INTERCOM will ask for each parameter in turn. If, after three attempts, you are still unsuccessful, INTERCOM will display a message asking you to get help. Before you attempt another LOGIN, you must go back to step 4 of the terminal hookup procedure. Once the LOGIN procedure is complete, the computer will signal that it is ready for further interaction

by printing:

COMMAND-

Executing the Program

The following commands will access QUEUE1 and begin its operation. Input each command in turn, exactly as written here and as is shown in the examples. After each individual command is transmitted with the carriage return key, wait for the "COMMAND-" response before entering the next command.

```
ATTACH,QUEUE1
REWIND,QUEUE1
FTN,I=QUEUE1,L=0
LGO
```

From this point on QUEUE1 will provide you with instructions. If the computer system is busy, there will be a short delay of as much as a minute for the computer to respond to inputs. The following examples demonstrate successful INTERCOM sessions with QUEUE1.

LOGOUT Procedure

When the program finishes, INTERCOM will revert to the COMMAND mode, signified by the computer generated "COMMAND-" printout. To terminate the session simply transmit the message

LOGOUT

After the computer prints its response to this command, push the TALK/CLEAR button on the DATA PHONE.

PROBLEM #1

You own a business that distributes wholesale merchandise throughout the state. You employ 25 drivers who deliver this merchandise from your warehouse to various retail stores. The drivers receive their loads and bills of lading from workers at the loading docks. You have employed 4 such workers. You wonder if you should fire one of these workers since there does not seem to be enough work to keep them busy most of the time. Each driver is paid $6.00 per hour. Each warehouse employee is paid $3.75 per hour. The following table shows how long it took to load 50 separate trucks.

Truck Number	Loading Time (hours)	Truck Number	Loading Time (hours)
1	.1	26	.4
2	.3	27	.1
3	.5	28	.1
4	.1	29	.1
5	.5	30	1.0
6	.4	31	.7
7	.2	32	.2
8	.3	33	.9
9	.7	34	.2
10	.6	35	.1
11	.2	36	.3
12	.5	37	1.2
13	.4	38	.5
14	.1	39	.5
15	1.1	40	.3
16	.2	41	.1
17	.8	42	.1
18	.3	43	.2
19	.1	44	.3
20	.6	45	.4
21	.4	46	.3
22	.6	47	.9
23	.2	48	.1
24	.8	49	.7
25	.1	50	.3

Also, you kept track of the time it took for your trucks to return, once they had left the warehouse to make a delivery. Fifty of those times are listed below.

Observation Number	Time	Observation Number	Time
1	.9	26	4.4
2	5.1	27	9.0
3	1.2	28	1.3
4	7.9	29	4.1
5	2.9	30	7.7
6	4.4	31	.5
7	2.9	32	5.0
8	6.1	33	2.6
9	.9	34	7.8
10	9.0	35	1.4
11	5.0	36	3.9
12	1.7	37	10.8
13	4.7	38	.4
14	11.4	39	5.8
15	1.5	40	3.0
16	8.9	41	8.0
17	2.6	42	1.1
18	3.0	43	4.9
19	1.7	44	5.0
20	.8	45	1.0
21	10.8	46	9.9
22	7.9	47	3.8
23	1.0	48	1.6
24	2.8	49	5.8
25	3.4	50	3.2

What should you do?

In this problem there are 4 servers and the customer population has a finite size of 25. Thus, the model used is the Finite Population M/M/C model with C equal to 4. The following printout represents the QUEUE1 solution to this problem. These explanations will help to define the output.

LAMBDA- Each driver returns, on the average, .23 times per hour; or the average round trip takes 4.41 hours.

LAMBDA1-5.16 drivers arrive for service every hour.

MU- On the average, each server can take care of 2.48 loads every hour.

RHO- LAMBDA/MU

P(0)- 11% of the time there are not trucks being loaded at the warehouse.

LS- At any given time, one could expect to see 2.21 trucks either being loaded or waiting to be loaded.

LQ- At any given time, .14 trucks will be waiting in line. Practically speaking, there is little or no waiting for service.

LB- Given that at least one loader (server) is already busy, .32 trucks could be expected to be waiting in line.

WS- On the average, a driver can expect to spend .43 hours from the time he arrives till he has been loaded and is ready to go again.

WQ- On the average, each driver can expect to spend .03 hours waiting to be loaded.

WB- If a driver knows that at least one truck is already being loaded, he can expect to wait .06 hours in the waiting line before being loaded.

CW- It cost the company $6.00 per hour to have drivers in the loading system instead of out on the road driving.

CS- Each loader (server) costs the company $3.75 per hour whether or not there is work for him to do.

C- Here, C = 3 is the most cost effective number of servers to employ for the loading operation.

Z- With 3 servers the average cost per hour for paying loaders and waiting drivers is $27.31

$$Z = C \times CS + LS \times CW$$

The conclusion reached is to fire one loader. A description

of the service system with only three servers can be generated by running QUEUE1 again.

```
ASD COMPUTER CENTER INTERCOM 4.7
SYSTEM  CSA
DATE  09/08/79        TIME  15.49.30.

PLEASE LOGIN
LOGIN,T720155,IPNG61,705


09/08/79   LOGGED IN AT  15.49.55.
           WITH USER-ID ED
           EQUIP/PORT 14/065
COMMAND- ATTACH,QUEUE1
 PFN IS
 QUEUE1
 PF CYCLE NO. = 001
COMMAND- REWIND,QUEUE1
COMMAND- FTN,I=QUEUE1,L=0
      2.907 CP SECONDS COMPILATION TIME
COMMAND- LGO

DO YOU WISH TO REVIEW THE INSTRUCTIONS FOR
THIS PROGRAM?  *NO
WHEN YOU ARE READY TO BEGIN, ENTER 'GO'.  *GO
THE MODELS AVAILABLE IN THIS PROGRAM ARE LISTED BELOW.

*MODEL*                                *CODE*
STANDARD M/M/1 MODEL                     1
FINITE QUEUE M/M/1 MODEL                 2
FINITE POPULATION M/M/1 MODEL            3
STANDARD M/M/C MODEL                     4
FINITE QUEUE M/M/C MODEL                 5
FINITE POPULATION M/M/C MODEL            6


DO YOU KNOW WHICH MODEL YOU WANT TO USE?  *NO

THE FOLLOWING SEQUENCE WILL HELP YOU CHOOSE THE APPROPRIATE MODEL.
HOW MANY SERVERS ARE THERE IN THE SERVICE SYSTEM?  *4
IS THE LENGTH OF THE QUEUE LIMITED TO A FINITE NUMBER?  *NO
IS THE POPULATION THAT MIGHT USE THE SERVICE FINITE
(USUALLY 30 UNITS OR LESS)?  *YES
ENTER THE SIZE OF THE FINITE POPULATION.  *25
FOR FINITE POPULATION PROBLEMS, CUSTOMER ARRIVAL DATA
SHOULD BE IN TERMS OF INTERARRIVAL TIMES (THE TIME
ELAPSED FROM WHEN A CUSTOMER LEAVES THE SERVICE
SYSTEM TILL THAT SAME CUSTOMER ARRIVES AGAIN FOR SERVICE.

DO YOU WISH TO ENTER SAMPLE DATA FOR INTERARRIVAL
TIMES?  *YES
```

```
YOU HAVE CHOSEN TO ENTER SAMPLE DATA.
LAMBDA, THE MEAN CUSTOMER ARRIVAL RATE, WILL BE
COMPUTED FOR YOU.  DATA WILL BE CHECKED TO
DETERMINE IF IT MEETS THE ASSUMPTIONS NECESSARY
FOR USE IN THESE MODELS.

ENTER THE NUMBER OF INTERARRIVAL TIMES OBSERVED.   •50

50 INTERARRIVAL OBSERVATION NUMBERS WILL BE
PRINTED.  AFTER EACH OBSERVATION NUMBER, ENTER
THE INTERARRIVAL TIME OBSERVED.

OBSERVATION #1   •.9
OBSERVATION #2   •5.1
OBSERVATION #3   •1.2
OBSERVATION #4   •7.9
OBSERVATION #5   •2.9
OBSERVATION #6   •4.4
OBSERVATION #7   •2.9
OBSERVATION #8   •6.1
OBSERVATION #9   •.9
OBSERVATION #10   •9.0
OBSERVATION #11   •5.0
OBSERVATION #12   •1.7
OBSERVATION #13   •4.7
OBSERVATION #14   •11.4
OBSERVATION #15   •1.5
OBSERVATION #16   •8.9
OBSERVATION #17   •2.6
OBSERVATION #18   •3.0
OBSERVATION #19   •1.7
OBSERVATION #20   •.8
OBSERVATION #21   •10.8
OBSERVATION #22   •7.9
OBSERVATION #23   •1.0
OBSERVATION #24   •2.8
OBSERVATION #25   •3.4
OBSERVATION #26   •4.4
OBSERVATION #27   •9.0
OBSERVATION #28   •1.3
OBSERVATION #29   •4.1
OBSERVATION #30   •7.7
OBSERVATION #31   •.5
OBSERVATION #32   •5.0
OBSERVATION #33   •2.6
OBSERVATION #34   •7.8
OBSERVATION #35   •1.4
OBSERVATION #36   •3.9
OBSERVATION #37   •10.8
OBSERVATION #38   •.4
```

```
OBSERVATION #39  •5.8
OBSERVATION #40  •[illegible].0
OBSERVATION #41  •8.0
OBSERVATION #42  •1.11
OBSERVATION #43  •4.9
OBSERVATION #44  •5.0
OBSERVATION #45  •1.0
OBSERVATION #46  •9.9
OBSERVATION #47  •3.2
OBSERVATION #48  •1.6
OBSERVATION #49  •5.8
OBSERVATION #50  •3.2
••••••••••••••••••••••••••••••••••••••••••••••••••••
BASED ON THE SAMPLE DATA PRESENTED, YOUR
TIME DISTRIBUTION MEETS THE ASSUMPTIONS OF THESE MODELS
••••••••••••••••••••••••••••••••••••••••••••••••••••
DO YOU WISH TO ENTER SAMPLE DATA FOR SERVICE TIMES?  •YES

YOU HAVE CHOSEN TO ENTER SAMPLE DATA.
MU, THE MEAN SERVICE RATE WILL BE COMPUTED FOR YOU.
DATA WILL BE CHECKED TO DETERMINE IF IT MEETS THE
ASSUMPTIONS NECESSARY FOR USE IN THESE MODELS.

ENTER THE NUMBER OF CUSTOMERS SERVICED WHILE DATA
WAS GATHERED (MAXIMUM 100).50

50 CUSTOMER NUMBERS WILL BE PRINTED.
AFTER EACH CUSTOMER NUMBER, ENTER THE AMOUNT OF TIME
NEEDED TO SERVICE EACH CUSTOMER.


CUSTOMER #1  •.1
CUSTOMER #2  •.3
CUSTOMER #3  •.5
CUSTOMER #4  •.1
CUSTOMER #5  •.5
CUSTOMER #6  •.4
CUSTOMER #7  •.2
CUSTOMER #8  •.3
CUSTOMER #9  •.7
CUSTOMER #10  •.6
CUSTOMER #11  •.2
CUSTOMER #12  •.5
CUSTOMER #13  •.4
CUSTOMER #14  •.1
CUSTOMER #15  •1.1
CUSTOMER #16  •.2
CUSTOMER #17  •.8
CUSTOMER #18  •.3
CUSTOMER #19  •.1
CUSTOMER #20  •.6
```

```
CUSTOMER #21   ◆.4
CUSTOMER #22   ◆.6
CUSTOMER #23   ◆.2
CUSTOMER #24   ◆.8
CUSTOMER #25   ◆.1
CUSTOMER #26   ◆.4
CUSTOMER #27   ◆.1
CUSTOMER #28   ◆.1
CUSTOMER #29   ◆.1
CUSTOMER #30   ◆1.0
CUSTOMER #31   ◆.7
CUSTOMER #32   ◆.2
CUSTOMER #33   ◆.9
CUSTOMER #34   ◆.2
CUSTOMER #35   ◆.1
CUSTOMER #36   ◆.3
CUSTOMER #37   ◆1.2
CUSTOMER #38   ◆.5
CUSTOMER #39   ◆.5
CUSTOMER #40   ◆.3
CUSTOMER #41   ◆.1
CUSTOMER #42   ◆.1
CUSTOMER #43   ◆.2
CUSTOMER #44   ◆.3
CUSTOMER #45   ◆.4
CUSTOMER #46   ◆.3
CUSTOMER #47   ◆.9
CUSTOMER #48   ◆.1
CUSTOMER #49   ◆.7
CUSTOMER #50   ◆.3
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
BASED ON THE SAMPLE DATA PRESENTED, YOUR
TIME DISTRIBUTION MEETS THE ASSUMPTIONS OF THESE MODELS
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

****

```
YOU HAVE SELECTED THE FINITE POPULATION M/M/C MODEL.


THE FOLLOWING VALUES DESCRIBE THE STEADY-STATE
OPERATION OF THE QUEUEING SYSTEM.

MEAN ARRIVAL RATE PER CUSTOMER
      (ARRIVALS PER TIME PERIOD):
            LAMBDA = .2267470863

EFFECTIVE MEAN ARRIVAL RATE FOR THE SYSTEM
      (CUSTOMERS PER TIME PERIOD):
            LAMBDA1 = 5.165697128376

MEAN SERVICE RATE (UNITS PER TIME PERIOD):
            MU = 2.487562189055

TRAFFIC INTENSITY:
            RHO = .09115232869258

NUMBER OF UNITS IN FINITE POPULATION:
            M = 25

NUMBER OF SERVERS:
            C = 4

PROBABILITY OF THE SYSTEM BEING IDLE:
            P(0) = .1095099664263

STEADY-STATE MEAN NUMBER OF UNITS IN SYSTEM:
            LS = 2.218242524436

STEADY-STATE MEAN NUMBER OF UNITS IN QUEUE:
            LQ = .1416322788285

STEADY-STATE MEAN NUMBER OF UNITS IN QUEUE FOR BUSY SYSTEM:
            LB = .3185443540656

STEADY-STATE MEAN TIME IN SYSTEM:
            WS = .4294178441571

STEADY-STATE MEAN TIME IN QUEUE:
            WQ = .02741784415708

STEADY-STATE MEAN TIME IN QUEUE FOR BUSY SYSTEM:
            WB = .06166531760365
```

****

```
DO YOU WANT TO FIND THE OPTIMAL NUMBER OF SERVERS (C)
BASED ON THE COST OF SERVICE AND THE COST OF WAITING?  *YES

ENTER THE COST OF WAITING PER CUSTOMER PER UNIT TIME.  *$6.00

ENTER THE COST PER SERVER PER UNIT TIME.  *$3.75

*****************************************************

THE COST OF WAITING PER CUSTOMER PER UNIT TIME IS:
          CW = $6.

THE COST PER SERVER PER UNIT TIME IS:
          CS = $3.75


THE OPTIMAL NUMBER OF SERVERS FOR THIS SYSTEM IS:
          C = 3

THE COST OF THE SYSTEM IN DOLLARS PER TIME PERIOD IS:
          Z = $27.3134814890?
*****************************************************
THIS PROBLEM IS COMPLETE.  DO YOU HAVE OTHER PROBLEMS
FOR THIS PROGRAM TO SOLVE?  *NO

IT HAS BEEN NICE WORKING WITH YOU.  HAVE A NICE DAY.

      STOP  END OF PROGRAM
        .465 CP SECONDS EXECUTION TIME
COMMAND- LOGOUT
CPA      4.143 SEC.          3.375 ADJ.
IO      23.041 SEC.          6.880 ADJ.
CRUS                        12.517
CONNECT TIME    0 HRS.   18 MIN.
 09/09/79  LOGGED OUT AT 16.07.13.
```

PROBLEM #2

Your local bookmaker, Joe, deals strictly on a phone-in basis. He runs one phone on a 4 line rotary system. This means that he can take a bet on one line and up to three more betters can call his number and wait for him to answer. If all four lines are busy, the fifth caller will get a busy signal. On the average, three betters try to call Joe every minute. Joe boasts that he averages 30 seconds per bet. Whenever you call Joe what is the probability that you will get a busy signal? What assumptions have you made?

In this problem sample data for arrival rate and service rate are not available. It must be assumed that the probability distributions meet the model assumptions. There is only one server, Joe. The number of customers who can enter the system is limited to four since there are only four telephone lines that reach Joe. If he is taking a bet from a customer on one of these lines (Joe is servicing a customer) then the maximum length of the waiting line is three. The mean arrival rate is three customers per minute. If Joe can handle 1 bet every 30 seconds, his mean service rate is 2 per minute. The system then is a Finite Queue M/M/1 type with LAMBDA equal to 3 and MU equal to 2. In this case you are rejected from the waiting line if you dial Joe's number and get a busy signal. P(N) equals .3838,

so on the average you will get a busy signal 38% of the time.

```
ASD COMPUTER CENTER INTERCOM 4.7
SYSTEM  CSA
DATE  09/08/79        TIME  16.07.51.

PLEASE LOGIN
LOGIN
ENTER PROBLEM NUMBER-T720155
[redacted] ENTER PASSWORD-
ENTER 3-DIGIT TERMINAL ID-705


09/08/79   LOGGED IN AT  16.08.28.
           WITH USER-ID ED
           EQUIP/PORT 14/065
COMMAND- ATTACH,QUEUE1
 PFN IS
 QUEUE1
 PF CYCLE NO. = 001
COMMAND- REWIND,QUEUE1
COMMAND- FTN,I=QUEUE1,L=0
      2.889 CP SECONDS COMPILATION TIME
COMMAND- LGO

DO YOU WISH TO REVIEW THE INSTRUCTIONS FOR
THIS PROGRAM?  •NO
WHEN YOU ARE READY TO BEGIN, ENTER 'GO'.  •GO
THE MODELS AVAILABLE IN THIS PROGRAM ARE LISTED BELOW.

•MODEL•                                •CODE•
STANDARD M/M/1 MODEL                     1
FINITE QUEUE M/M/1 MODEL                 2
FINITE POPULATION M/M/1 MODEL            3
STANDARD M/M/C MODEL                     4
FINITE QUEUE M/M/C MODEL                 5
FINITE POPULATION M/M/C MODEL            6


DO YOU KNOW WHICH MODEL YOU WANT TO USE?  •NO

THE FOLLOWING SEQUENCE WILL HELP YOU CHOOSE THE APPROPRIATE MODEL.
HOW MANY SERVERS ARE THERE IN THE SERVICE SYSTEM?  •1
IS THE LENGTH OF THE QUEUE LIMITED TO A FINITE NUMBER?  •YES
ENTER THE MAXIMUM LENGTH OF THE QUEUE.  •3

DO YOU WISH TO ENTER SAMPLE DATA
FOR CUSTOMER ARRIVALS? •NO
ENTER LAMBDA, THE MEAN (AVERAGE) ARRIVAL RATE
(UNITS PER TIME PERIOD).  •3
```

```
DO YOU WISH TO ENTER SAMPLE DATA FOR SERVICE TIMES?  •NO
ENTER MU, THE MEAN (AVERAGE) SERVICE RATE (UNITS
PER TIME PERIOD.  •2

***************************************************

YOU HAVE SELECTED THE FINITE QUEUE M/M/1 MODEL.


THE FOLLOWING VALUES DESCRIBE THE STEADY-STATE
OPERATION OF THE QUEUEING SYSTEM.

MEAN ARRIVAL RATE (UNITS PER TIME PERIOD):
          LAMBDA = 3.

MEAN SERVICE RATE (UNITS PER TIME PERIOD):
          MU = 2.

TRAFFIC INTENSITY:
          RHO = 1.5

MAXIMUM NUMBER ALLOWED IN THE SYSTEM:
          N = 4

PROBABILITY OF REJECTION DUE TO A FULL QUEUE:
          P(N) = .3838862559242

NUMBER OF SERVERS:
          C = 1

PROBABILITY OF THE SYSTEM BEING IDLE:
          P(0) = .0758293838625

STEADY-STATE MEAN NUMBER OF UNITS IN SYSTEM:
          LS = 2.758293838863

STEADY-STATE MEAN NUMBER OF UNITS IN QUEUE:
          LQ = 1.834123222749

STEADY-STATE MEAN NUMBER OF UNITS IN QUEUE FOR BUSY SYSTEM:
          LB = 1.984615384615

STEADY-STATE MEAN TIME IN SYSTEM:
          WS = 1.492307692308

STEADY-STATE MEAN TIME IN QUEUE:
          WQ = .9923076923077

STEADY-STATE MEAN TIME IN QUEUE FOR BUSY SYSTEM:
          WB = 1.073727810651
```

```
*****************************************************

DO YOU WANT TO FIND THE OPTIMAL SERVICE RATE (MU)
BASED ON THE COST OF SERVICE AND THE COST OF WAITING?  *NO

THIS PROBLEM IS COMPLETE.  DO YOU HAVE OTHER PROBLEMS
FOR THIS PROGRAM TO SOLVE?  *NO

IT HAS BEEN NICE WORKING WITH YOU.  HAVE A NICE DAY.

      STOP  END OF PROGRAM
        .109 CP SECONDS EXECUTION TIME
COMMAND- LOGOUT
CPA        3.388 SEC.           2.761 ADJ.
IO        21.452 SEC.           6.349 ADJ.
CPUS                           11.276
CONNECT TIME     0 HRS.     5 MIN.
 09/08/79  LOGGED OUT AT 16.13.16.
```

APPENDIX D

SURVEY QUESTIONS

The following information will be used to evaluate the effectiveness of QUEUE1 and the instructions you were given. Please make any comments, positive or negative, you feel would aid this evaluation.

1. What is your profession?

2. What is your highest level of education?

3. List any degrees you have earned.

4. What is the extent of your knowledge of queueing theory?

5. List your level of experience with computers.

6. How familiar are you with remote computer terminals?

7. List any problems you had with the following:

    a. terminal controls/keyboard.

    b. LOGIN procedure.

    c. accessing QUEUE1.

    d. data input.

    e. selecting the correct model.

f. LOGOUT procedure.

g. interpreting the results of the exercise.

8. Comment on the printed instructions:

   a. length.

   b. detail.

   c. clarity.

   d. accuracy.

9. Indicate how confident you are that QUEUE1 can aid decision-making in real situations.

1	2	3	4	5
least confident				most confident

10. Additional comments.

VITA

Paul Hamilton was born on 1 September 1948 in Philadelphia, Pennsylvania. He received a Bachelor of Science degree from the United States Air Force Academy in June 1971. He attended Undergraduate Pilot Training at Vance Air Force Base, Oklahoma from September 1971 to July 1972. He served as a C-130 pilot at Pope Air Force Base, North Carolina until entering the School of Engineering at the Air Force Institute of Technology in May 1978.

Permanent address: 3416 Oak Grove
Midwest City, Oklahoma 73110